

# Chương 4: Tầng Mạng

## Mục tiêu

- ❑ Nguyên tắc triển khai dịch vụ của tầng Mạng:
  - Định tuyến (Lựa chọn đường đi)
  - Mở rộng Phạm vi
  - Cách thức router hoạt động
  - Nâng cao: IPv6, multicast
- ❑ Cài đặt những dịch vụ này trên Internet như thế nào

## Học cái gì:

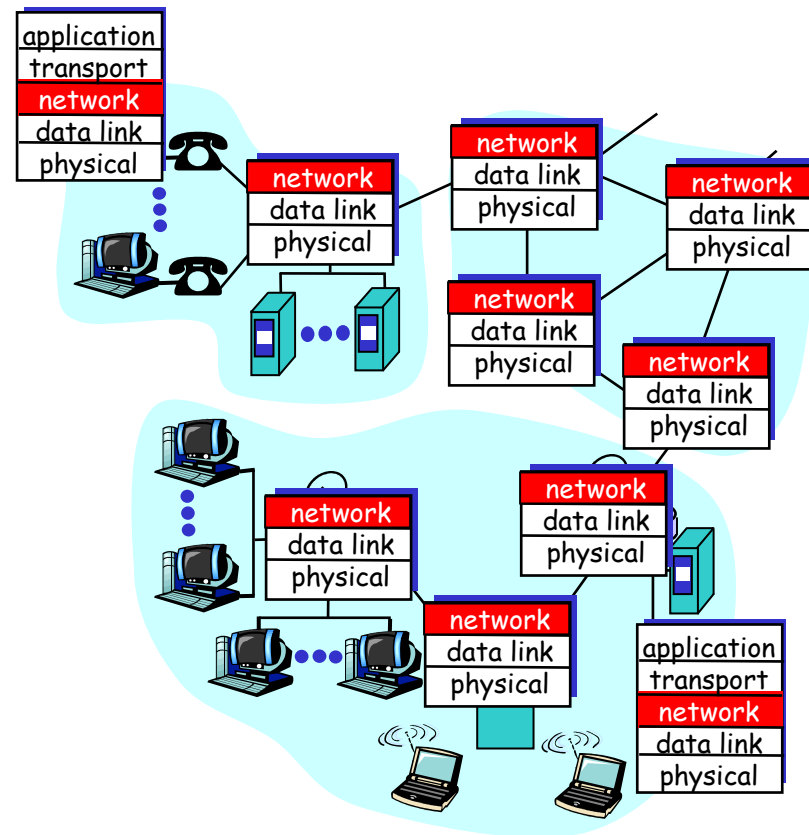
- ❑ Các dịch vụ của tầng Mạng
- ❑ Nguyên tắc định tuyến: lựa chọn đường đi
- ❑ Định tuyến phân cấp
- ❑ Giao thức IP
- ❑ Giao thức định tuyến trên Internet
  - Nội miền
  - Liên miền
- ❑ Kiến trúc Router
- ❑ IPv6
- ❑ NAT, PAT
- ❑ multicast routing

# Chức năng của tầng Mạng

- ❑ Chuyển gói tin từ máy tính gửi đến máy tính nhận
- ❑ Giao thức tầng Mạng có mặt trên *tất cả* máy tính, router

## Ba chức năng cơ bản:

- ❑ **Lựa chọn tuyến đường:** Tuyến đường gói tin đi từ nguồn đến đích. *Thuật toán định tuyến*
- ❑ **Chuyển mạch:** Router chuyển gói tin từ một đầu vào này ra đầu ra thích hợp
- ❑ **Thiết lập tuyến đường:** một số kiến trúc mạng đòi hỏi tuyến đường gói tin đi qua phải được thiết lập trước



# Mô hình Dịch vụ tầng Mạng

?: Chọn *Mô hình dịch vụ* nào của “kênh truyền” để chuyển gói dữ liệu từ nơi gửi đến nơi nhận ?

Trình tượng hóa dịch vụ

- ☐ Đảm bảo băng thông không?
- ☐ Đảm bảo tốc độ nhận đều không (không có hiện tượng jitter)?
- ☐ Không mất mát ?
- ☐ Đảm bảo thứ tự?
- ☐ Có thông báo về tình trạng tắc nghẽn cho bên gửi không ?

**Vấn đề** quan trọng nhất mà tầng Mạng phải trả lời :

**Mạch ảo  
hay  
Mạch gói?**

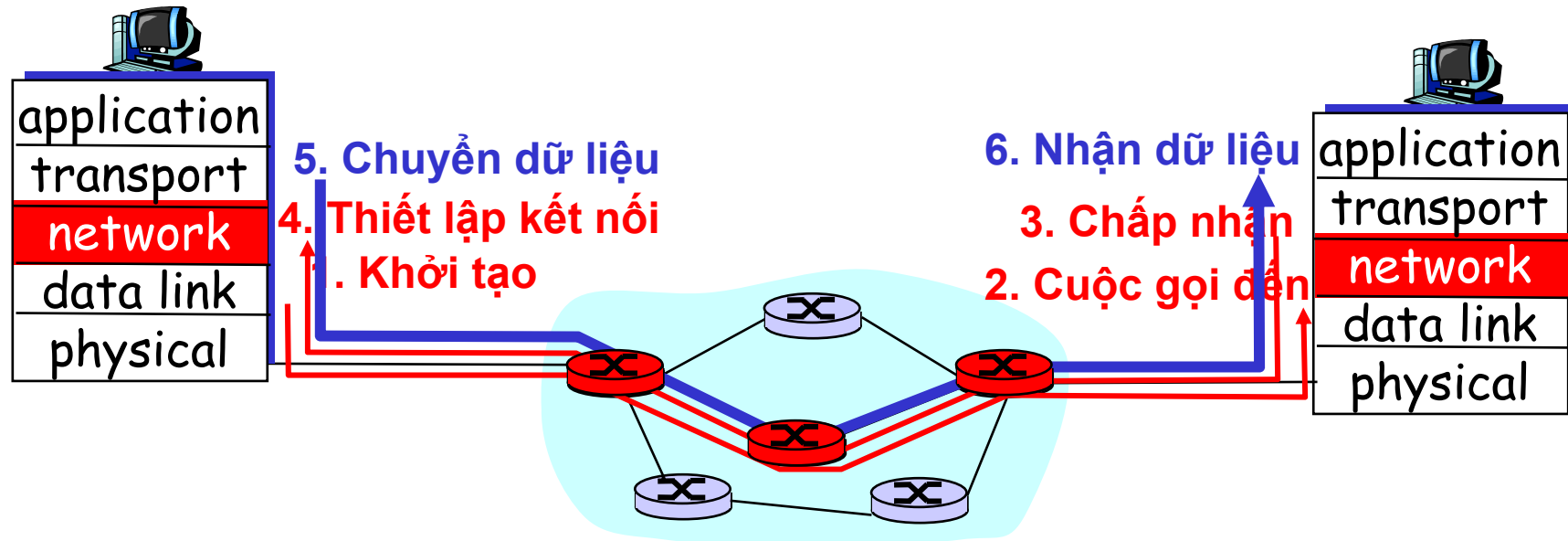
# Chuyển mạch ảo

Tuyến đường từ đích đến nguồn giống như mạch trong mạng điện thoại

- Quan trọng : Hiệu suất
  - Thiết lập kết nối dọc theo tuyến đường
- 
- ❑ Thiết lập và Đóng kết nối *trước* khi truyền dữ liệu
  - ❑ Mỗi packet mang định danh mạch ảo là VC identifier (Không phải địa chỉ máy đích)
  - ❑ *Tất cả* router trên tuyến đường duy trì trạng thái của kết nối đi qua
    - Kênh truyền ở tầng giao vận chỉ liên quan đến hai thực thể đầu cuối
  - ❑ Tài nguyên của đường truyền (Băng thông, Bộ đệm) phải được cấp phát cho mỗi mạch ảo
    - Để đạt được hiệu suất như mạng điện thoại.

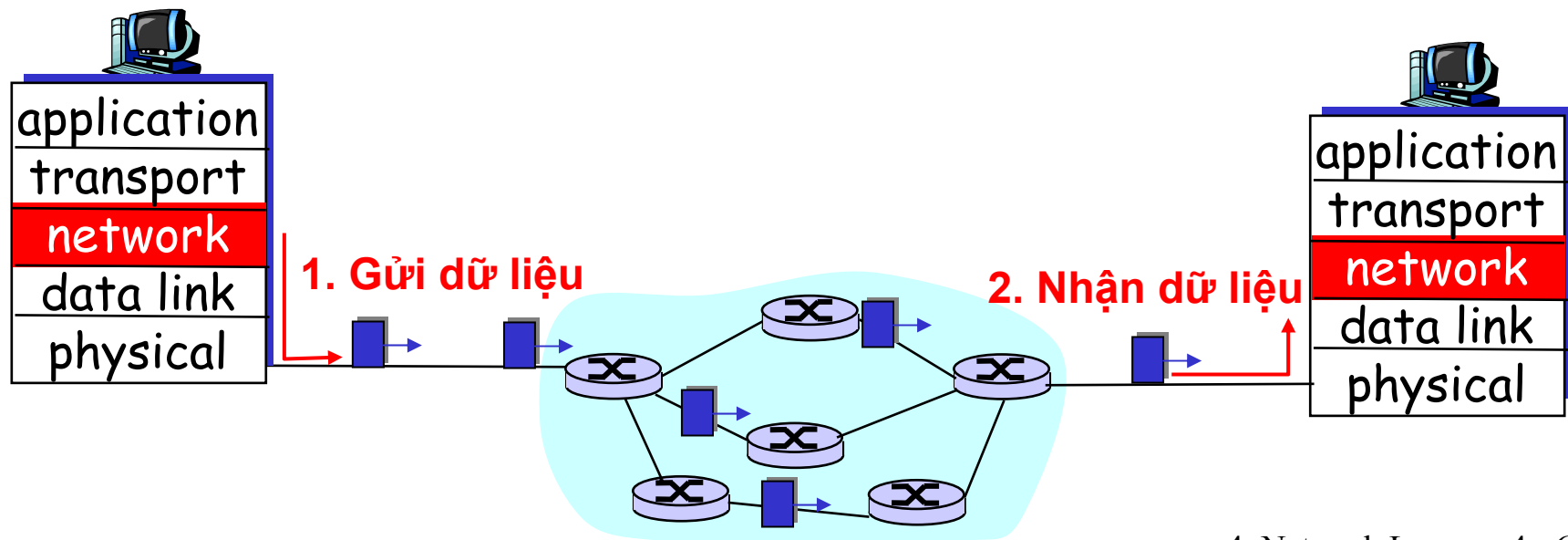
# Mạch ảo: Giao thức báo hiệu (Signal)

- ❑ Để Thiết lập và Đóng mạch ảo (VC)
- ❑ Có trong ATM, Frame-relay, X.25
- ❑ Không được sử dụng trên Internet ngày nay



# Mạng chuyển mạch gói: Mô hình Internet

- ❑ Không cần thiết lập đường truyền ở tầng Mạng
- ❑ routers: Không duy trì trạng thái các kết nối đi qua
  - no network-level concept of “connection”
- ❑ Gói tin được định tuyến dựa trên địa chỉ gói nhận
  - Các gói tin có thể đi theo những tuyến đường khác nhau



# Mô hình Dịch vụ ở tầng Mạng

| Kiến trúc<br>Mạng | Mô hình<br>Dịch vụ | Có đảm bảo ?       |     |        |           | Phản hồi<br>Tắc nghẽn  |
|-------------------|--------------------|--------------------|-----|--------|-----------|------------------------|
|                   |                    | Băng thông         | Mất | Thứ tự | Thời gian |                        |
| Internet          | best effort        | none               | no  | no     | no        | no (inferred via loss) |
| ATM               | CBR                | constant rate      | yes | yes    | yes       | no congestion          |
| ATM               | VBR                | guaranteed rate    | yes | yes    | yes       | no congestion          |
| ATM               | ABR                | guaranteed minimum | no  | yes    | no        | yes                    |
| ATM               | UBR                | none               | no  | yes    | no        | no                     |

- ❑ Mô hình dịch vụ trên Internet : Intserv, Diffserv
  - Chương 6

# Chuyển Mạch Ảo hay Chuyển Mạch Gói?

## Internet

- ❑ Dữ liệu trao đổi giữa máy tính
  - Dịch vụ mạng tính “co giãn”, không đòi hỏi chặt chẽ thời gian.
- ❑ Thiết bị đầu cuối “thông minh”
  - Có thể thích nghi, Kiểm soát, Khắc phục lỗi
  - “Lỗi” mạng đơn giản, Phức tạp đặt ở “Rìa”
- ❑ Nhiều kiểu môi trường truyền dẫn
  - Các đặc điểm khác nhau
  - Không thể có chung chất lượng dịch vụ

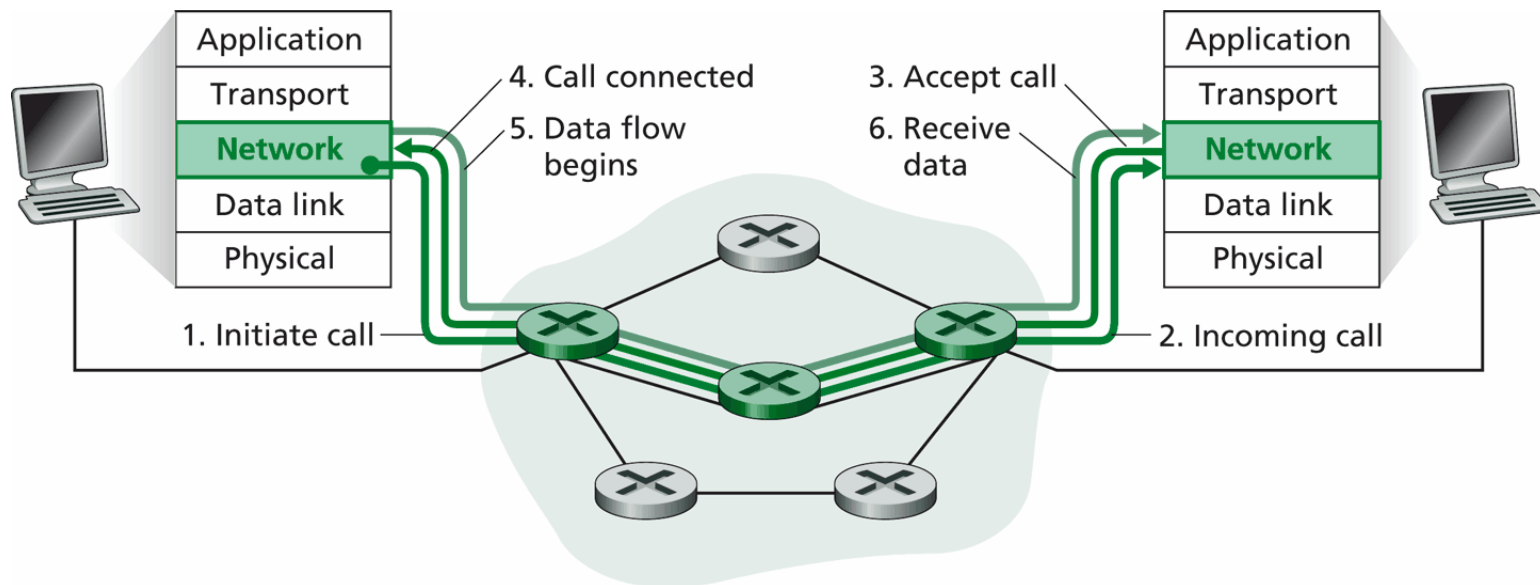
## ATM

- ❑ Phát triển từ Mạng điện thoại
- ❑ Hội thoại của con người:
  - Yêu cầu nghiêm ngặt về Thời gian và Độ tin cậy
  - Đảm bảo chất lượng dịch vụ
- ❑ Thiết bị đầu cuối đơn giản
  - Máy điện thoại
  - Phức tạp đặt ở bên trong Mạng

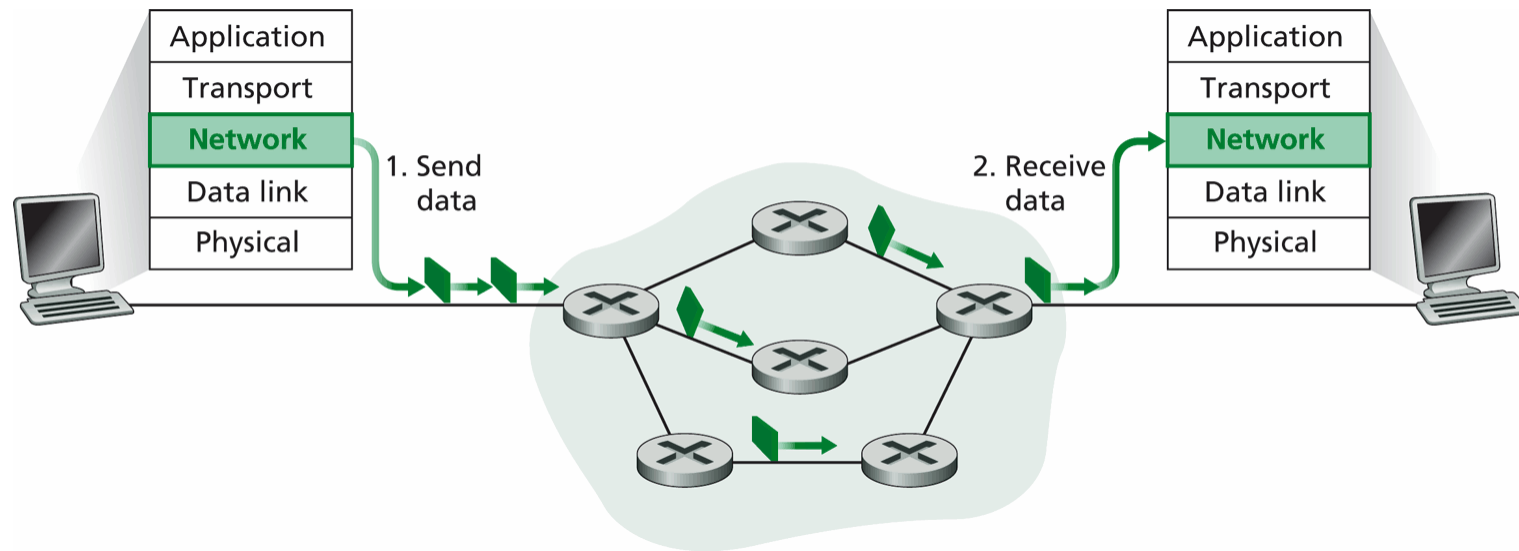


# Sử dụng Mạng ảo để cài đặt Dịch vụ ở tầng Mạng

- Để cung cấp thêm một số chức năng, có thể lựa chọn công nghệ mạng ảo – ví dụ Virtual Private Network (VPN)



# Chuyển mạch gói : Cài đặt các chức năng Mạng cơ bản nhất

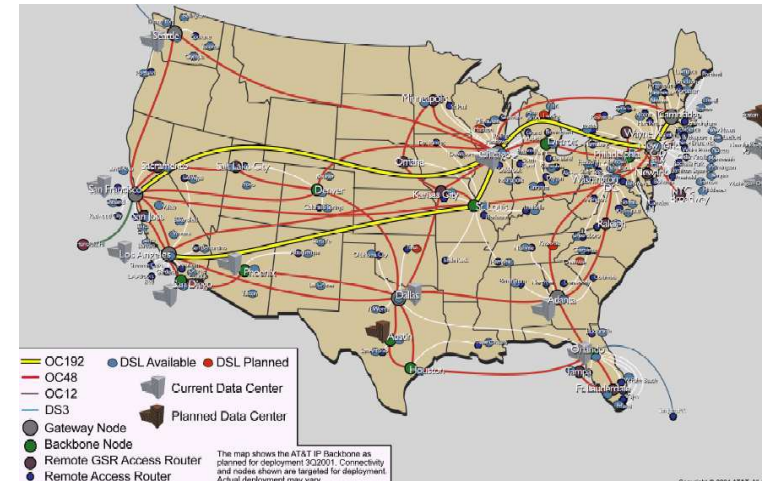


- ❑ Thường chỉ cung cấp dịch vụ cơ bản nhất: chuyển gói tin đi từ nơi Gửi đến nơi Nhận.
- ❑ Tập trung vào mô hình Internet theo kiểu “cố gắng tối đa” này

# Định tuyến

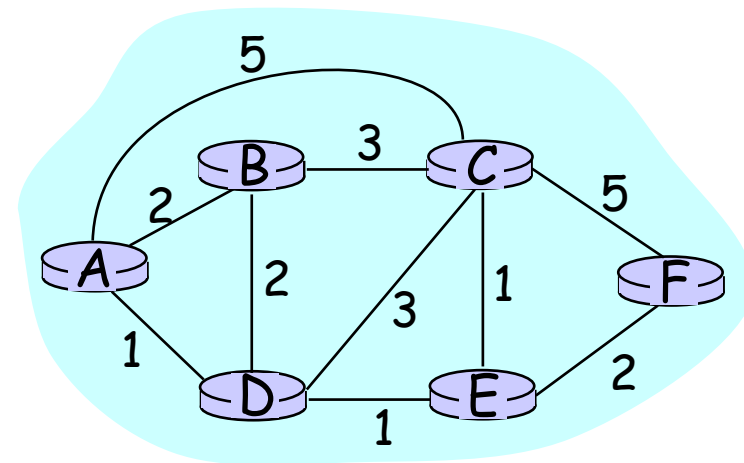
## Định tuyến

**Mục tiêu:** xác định tuyến đường “tốt” (dãy các router) trên mạng từ nút gửi đến nút nhận.

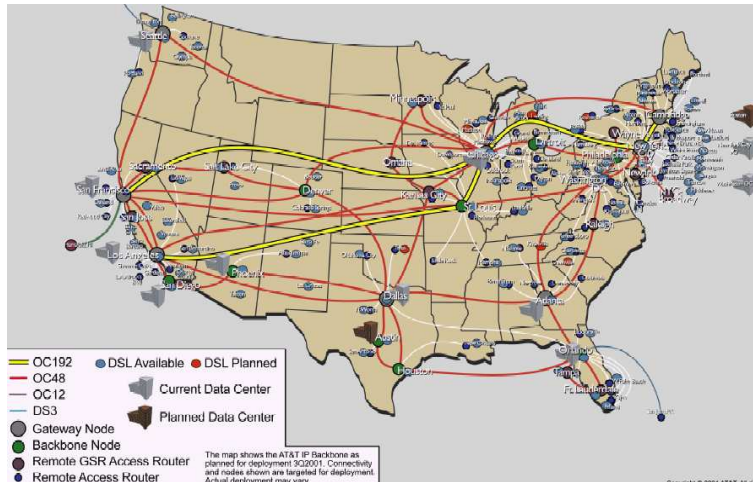


Đồ thị để xây dựng thuật toán định tuyến:

- Định là Router
- Cạnh là các đường kết nối trực tiếp
  - Giá của cạnh: Độ trễ, Chi phí, hay Mức độ Tắc nghẽn



# Định tuyến : Các Yêu cầu khác



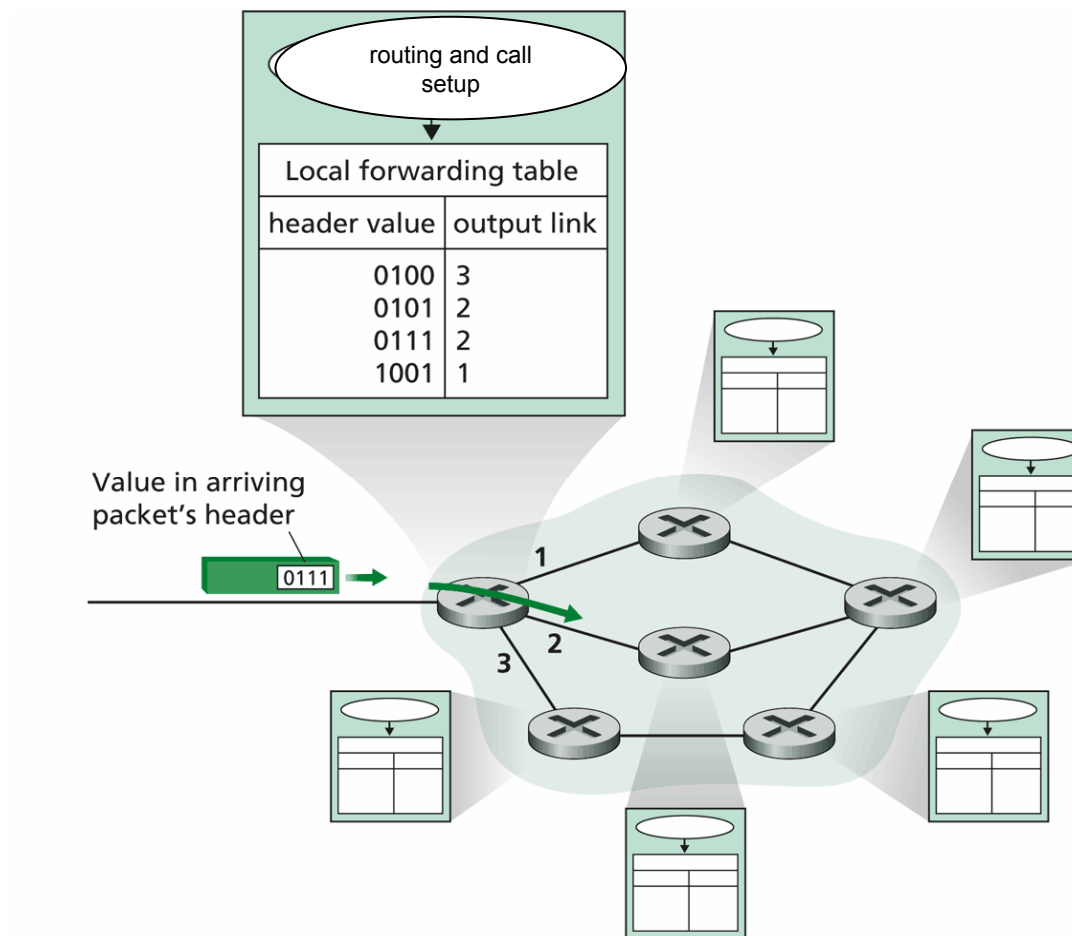
## □ Một vài Yêu cầu khác

- Mạnh và Tin cậy
- Tối ưu và có Hiệu quả
- Phân tán, Tự quản, Không có điểm hỏng duy nhất
- Đơn giản
- Công bằng

### Định tuyến

**Mục tiêu:** xác định tuyến đường “tốt” (dãy các router) trên mạng từ nút gửi đến nút nhận.

# Minh họa Chức năng Tầng Mạng



# Không gian Định tuyến

## □ Định tuyến

- Ai quyết định tuyến đường?
  - Nút gửi quyết định
  - Mạng quyết định
- Có bao nhiêu tuyến đường từ nút Gửi s đến nút Nhận d?
  - Định tuyến nhiều chặng
  - Định tuyến nhiều chặng
- Tuyến đường có thay đổi theo tải của mạng không?
  - Định tuyến thích nghi
  - Định tuyến tĩnh
- ...

# Phân loại Thuật toán Định tuyến

**Thông tin *Toàn cục* hay  
*Phân tán* ?**

**Toàn cục (Global):**

- ❑ Mọi router đều biết về toàn bộ topo của đồ thị
- ❑ Thuật toán “link state”

**Phân tán (Decentralized):**

- ❑ Mỗi router chỉ biết về router và giá đến các router hàng xóm
- ❑ Thực hiện quá trình trao đổi thông tin với các “hàng xóm”
- ❑ Thuật toán “distance vector”

**Động hay Tĩnh?**

**Tĩnh (Static):**

- ❑ Router rất ít thay đổi

**Động (Dynamic):**

- ❑ Router thay đổi thường xuyên
  - Cập nhật định kỳ
  - Chạy lại thuật toán khi có 1 giá đường đi thay đổi

# Thuật toán Định tuyến Link-State

## Thuật toán Dijkstra

- ❑ Tất cả các nút đều biết được về topo của toàn mạng
  - Biết được do các thông điệp quảng cáo được gửi quảng bá
  - Tất cả các nút có thông tin giống nhau
- ❑ Tính toán đường đi tốt nhất đến tất cả các nút khác
  - Tạo ra **Bảng Định tuyến**
- ❑ Sau k bước tính toán, xác định được đường ngắn nhất tới k đích

## Ký hiệu:

- ❑  $c(i,j)$ : Giá đường đi từ i tới j. Có giá trị vô cùng nếu i và j không có đường trực tiếp
- ❑  $D(v)$ : Giá hiện tại của đường đi tới đích V
- ❑  $p(v)$ : Nút kề trước V trong tuyến đường đến V
- ❑  $N$ : Tập hợp các đỉnh đã xác định được đường đi ngắn nhất



# Thuật toán Dijkstra

1 **Khởi tạo:**

2  $N = \{A\}$

3 Với tất cả các nút  $v$

4 Nếu  $v$  kề với  $A$

5 thì  $D(v) = c(A, v)$

6 nếu không  $D(v) = \text{infty}$

7

8 **Lặp**

9 Tìm  $w$  không trong  $N$  sao cho  $D(w)$  nhỏ nhất

10 Bổ sung  $w$  vào  $N$

11 Cập nhật  $D(v)$  cho tất cả  $v$  kề với  $w$  và không trong  $N$ :

12  $D(v) = \min( D(v), D(w) + c(w, v) )$

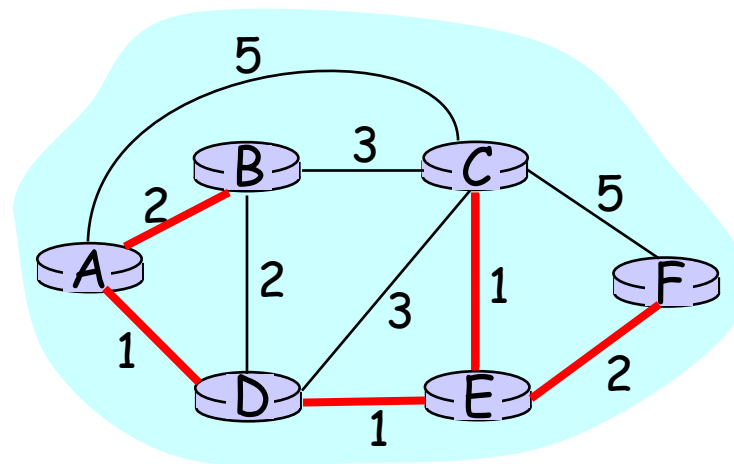
13 /\* Giá mới tới  $v$  hoặc là giá cũ đến  $v$  hoặc đường đi ngắn nhất

14 đến  $w$  cộng thêm khoảng cách từ  $w$  tới  $v$  \*/

15 **Cho đến khi tất cả các nút đều nằm trong  $N$**

# Ví dụ về Thuật toán Dijkstra

| Bước | Tập N  | D(B),p(B) | D(C),p(C) | D(D),p(D) | D(E),p(E) | D(F),p(F) |
|------|--------|-----------|-----------|-----------|-----------|-----------|
| → 0  | A      | 2,A       | 5,A       | 1,A       | vô cùng   | vô cùng   |
| → 1  | AD     | 2,A       | 4,D       |           | 2,D       | vô cùng   |
| → 2  | ADE    | 2,A       | 3,E       |           |           | 4,E       |
| → 3  | ADEB   |           | 3,E       |           |           | 4,E       |
| → 4  | ADEBC  |           |           |           |           | 4,E       |
| 5    | ADEBCF |           |           |           |           |           |



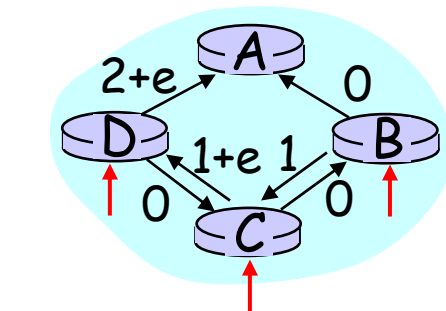
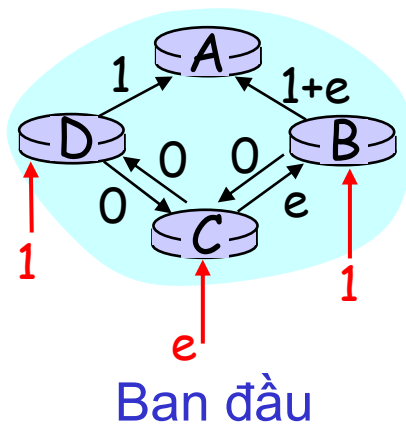
# Thảo luận về Thuật toán Dijkstra

**Độ phức tạp Thuật toán:** n nút

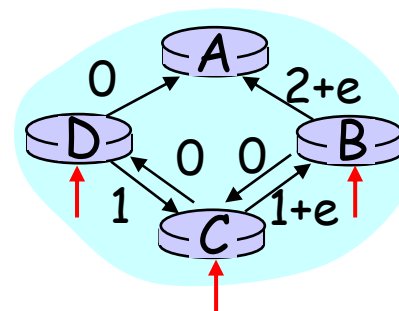
- ❑ Mỗi vòng lặp: Cần kiểm tra tất cả các nút w không ở trong N
- ❑  $n*(n+1)/2 : O(n^2)$
- ❑ Bằng thuật toán “mịn” hơn:  $O(n \log n)$

**Tình huống Dao động:**

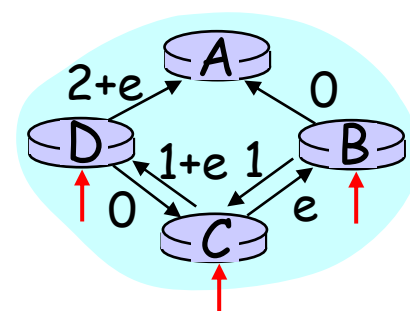
- ❑ Ví dụ khi Chi phí kênh truyền = khối lượng dữ liệu truyền qua



... Tính lại Định tuyến



... Tính lại



... Tính lại

# Thuật toán Định tuyến Distance Vector

## Lặp:

- ❑ Liên tục diễn ra cho đến khi không còn thông điệp trao đổi.
- ❑ *Tự kết thúc*: Không có “tín hiệu” để dừng lại

## Không đồng bộ:

- ❑ Các nút sau khi gửi/ nhận thông điệp không bị phong tỏa!

## Phân tán:

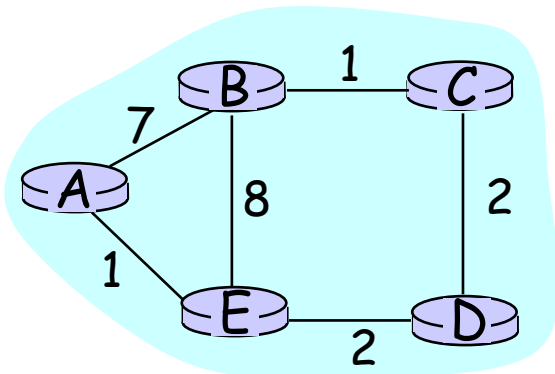
- ❑ Mỗi nút chỉ truyền thông với hàng xóm

## Cấu trúc Dữ liệu Bảng Distance

- ❑ Mỗi nút có một Bảng riêng
- ❑ Mỗi hàng ứng với một đích cụ thể
- ❑ Mỗi cột ứng với một hàng xóm có đường kết nối trực tiếp
- ❑ Ví dụ: trong nút X, với đích Y qua hàng xóm Z:

$$\begin{aligned} D^X(Y, Z) &= \text{Khoảng cách từ X tới Y, qua Z là chặng tiếp} \\ &= c(X, Z) + \min_w \{D^Z(Y, w)\} \end{aligned}$$

# Ví dụ Bảng Distance



$$D^E(C,D) = c(E,D) + \min_w \{D^D(C,w)\} \\ = 2+2 = 4$$

$$D^E(A,D) = c(E,D) + \min_w \{D^D(A,w)\} \\ = 2+3 = 5$$

$$D^E(A,B) = c(E,B) + \min_w \{D^B(A,w)\} \\ = 8+6 = 14$$

Chi phí đến các đích thông qua

| $D^E()$ | A | B  | D |
|---------|---|----|---|
| A       | 1 | 14 | 5 |
| B       | 7 | 8  | 5 |
| C       | 6 | 9  | 4 |
| D       | 4 | 11 | 2 |

Đích

# Bảng Distance tạo ra Bảng định tuyến

Chí phí đến đích qua

| $D^E()$ | A | B  | D |
|---------|---|----|---|
| A       | 1 | 14 | 5 |
| B       | 7 | 8  | 5 |
| C       | 6 | 9  | 4 |
| D       | 4 | 11 | 2 |

Đích

Đường ra được dùng + Giá

|   |     |
|---|-----|
| A | A,1 |
| B | D,5 |
| C | D,4 |
| D | D,2 |

Đích

Bảng Distance → Bảng định tuyến

# Tổng quan Định tuyến Distance Vector

**Lặp, Không đồng bộ:** mỗi lần tính toán cục bộ là do:

- ❑ Chi phí một đường thay đổi
- ❑ Thông điệp từ hàng xóm: Đường đi ngắn nhất của hàng xóm cũng thay đổi

**Phân tán:**

- ❑ Mỗi nút chỉ thông báo với hàng xóm *khi* có một giá đường đi nào đó thay đổi
  - Chỉ thông báo trong trường hợp cần phải thông báo

**Mỗi nút**

*Đợi* for (Thông báo giá đường đi thay đổi từ Hàng xóm)

*Tính lại* bằng distance

Nếu có giá thay đổi, *Thông báo* với các hàng xóm

# Thuật toán Distance Vector

At all nodes, X:

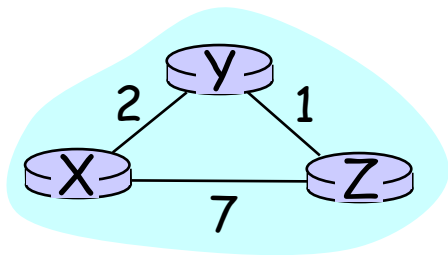
- 1 Initialization:
- 2 for all adjacent nodes v:
- 3      $D^X(*,v) = \text{infty}$      /\* the \* operator means "for all rows" \*/
- 4      $D^X(v,v) = c(X,v)$
- 5 for all destinations, y
- 6     send  $\min_w D^X(y,w)$  to each neighbor /\* w over all X's neighbors \*/



# Thuật toán Distance Vector:

```
8 loop
9   wait (until I see a link cost change to neighbor V
10      or until I receive update from neighbor V)
11
12   if (c(X,V) changes by d)
13     /* change cost to all dest's via neighbor v by d */
14     /* note: d could be positive or negative */
15     for all destinations y:  $D^X(y,V) = D^X(y,V) + d$ 
16
17   else if (update received from V wrt destination Y)
18     /* shortest path from V to some Y has changed */
19     /* V has sent a new value for its  $\min_w DV(Y,w)$  */
20     /* call this received new value is "newval" */
21     for the single destination y:  $D^X(Y,V) = c(X,V) + \text{newval}$ 
22
23   if we have a new  $\min_w D^X(Y,w)$  for any destination Y
24     send new value of  $\min_w D^X(Y,w)$  to all neighbors
25
26 forever
```

# Ví dụ Thuật toán Distance Vector



|             |   | cost via |          |
|-------------|---|----------|----------|
|             |   | Y        | Z        |
| destination | X |          |          |
|             | Y | 2        | $\infty$ |
|             | Z | $\infty$ | 7        |

|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | Y        | Z |
| destination | X |          |   |
|             | Y | 2        | 8 |
|             | Z | 3        | 7 |

|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | Y        | Z |
| destination | X |          |   |
|             | Y |          |   |
|             | Z |          |   |

|             |   | cost via |          |
|-------------|---|----------|----------|
|             |   | X        | Z        |
| destination | X | 2        | $\infty$ |
|             | Y | $\infty$ | 1        |
|             | Z | $\infty$ | 1        |

|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | X        | Z |
| destination | X | 2        | 8 |
|             | Y | 9        | 1 |
|             | Z |          |   |

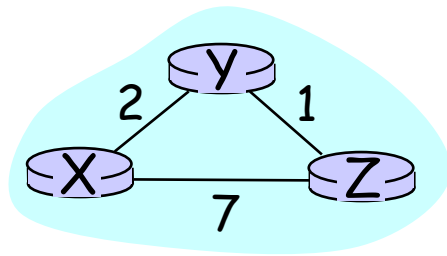
|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | X        | Z |
| destination | X |          |   |
|             | Y |          |   |
|             | Z |          |   |

|             |   | cost via |          |
|-------------|---|----------|----------|
|             |   | X        | Y        |
| destination | X | 7        | $\infty$ |
|             | Y | $\infty$ | 1        |
|             | Z |          |          |

|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | X        | Y |
| destination | X | 7        | 3 |
|             | Y | 9        | 1 |
|             | Z |          |   |

|             |   | cost via |   |
|-------------|---|----------|---|
|             |   | X        | Y |
| destination | X |          |   |
|             | Y |          |   |
|             | Z |          |   |

# Ví dụ Thuật toán Distance Vector



|                  |   | cost via |          |
|------------------|---|----------|----------|
|                  |   | Y        | Z        |
| d<br>e<br>s<br>t | X | 2        | $\infty$ |
|                  | Z | $\infty$ | 7        |

|                  |   | cost via |          |
|------------------|---|----------|----------|
|                  |   | X        | Z        |
| d<br>e<br>s<br>t | Y | 2        | $\infty$ |
|                  | Z | $\infty$ | 1        |

|                  |   | cost via |          |
|------------------|---|----------|----------|
|                  |   | X        | Y        |
| d<br>e<br>s<br>t | Z | 7        | $\infty$ |
|                  | Y | $\infty$ | 1        |

|                  |   | cost via |   |
|------------------|---|----------|---|
|                  |   | Y        | Z |
| d<br>e<br>s<br>t | X | 2        | 8 |
|                  | Z | 3        | 7 |

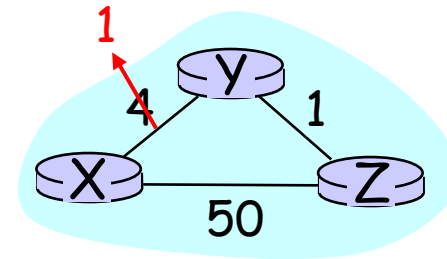
$$D^X(Y,Z) = c(X,Z) + \min_w \{D^Z(Y,w)\} \\ = 7 + 1 = 8$$

$$D^X(Z,Y) = c(X,Y) + \min_w \{D^Y(Z,w)\} \\ = 2 + 1 = 3$$

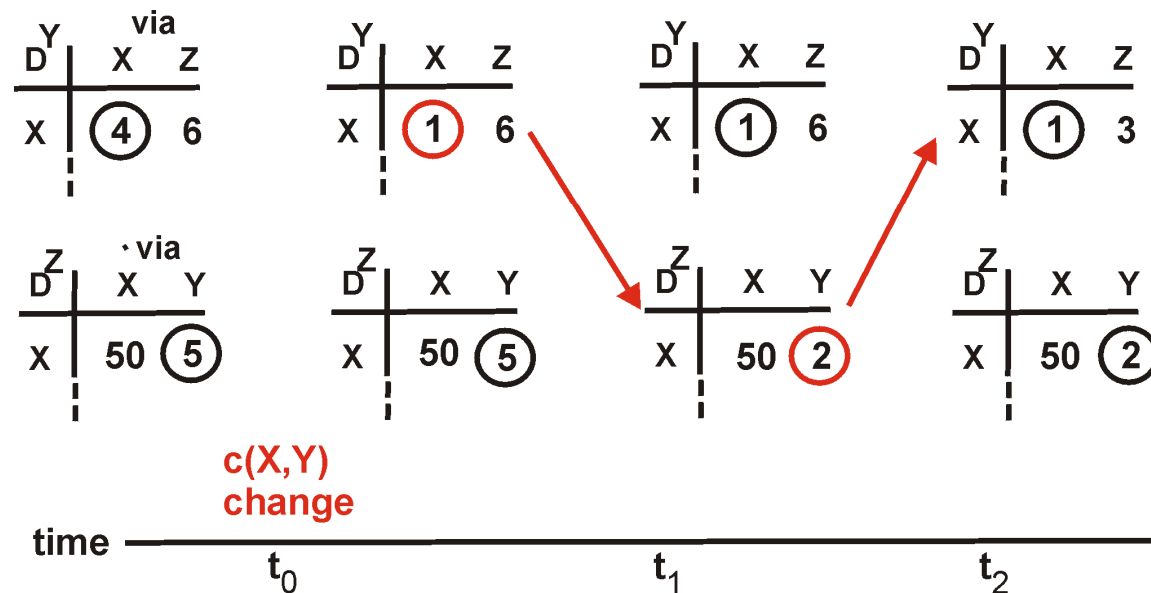
# Distance Vector: Thay đổi giá đường đi

## Thay đổi giá đường đi:

- ❑ Nút phát hiện giá đường đi thay đổi
- ❑ Cập nhật Bảng Distance (line 15)
- ❑ Nếu có một giá nào đó thay đổi, thông báo với các hàng xóm (dòng 23,24)



“Tin  
tốt  
lan  
nhANH”

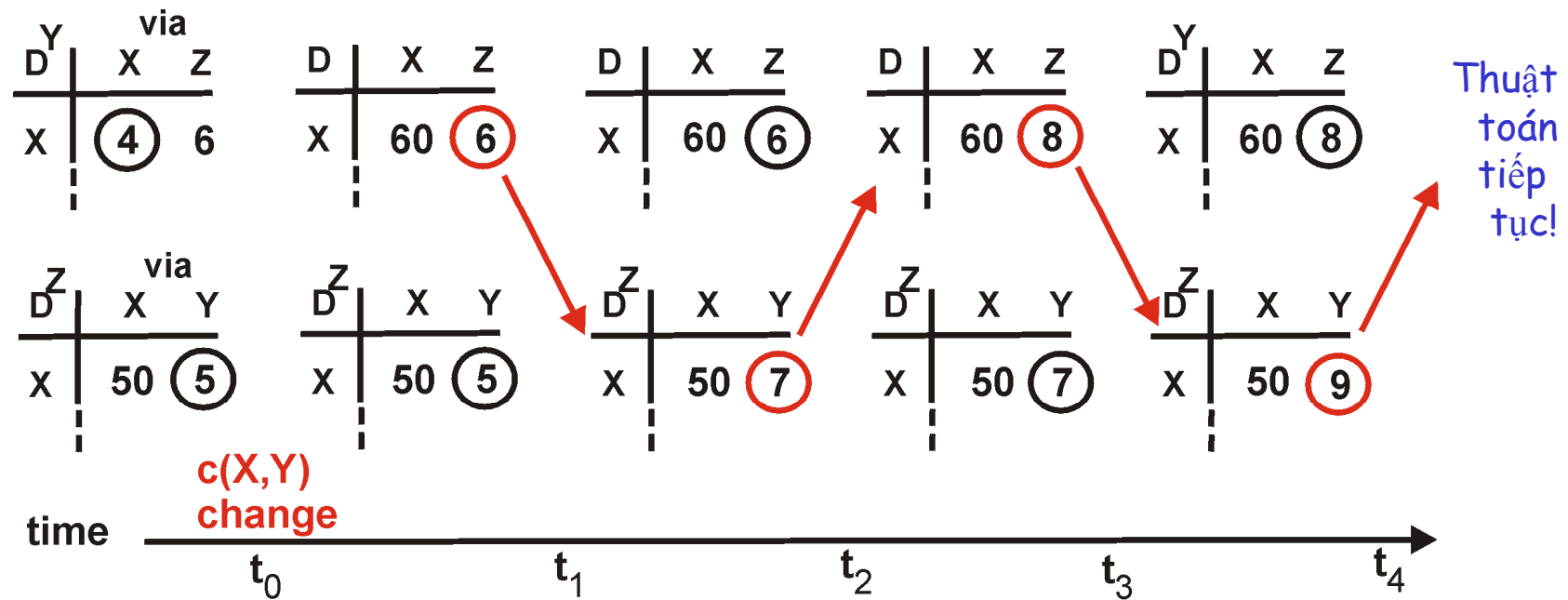
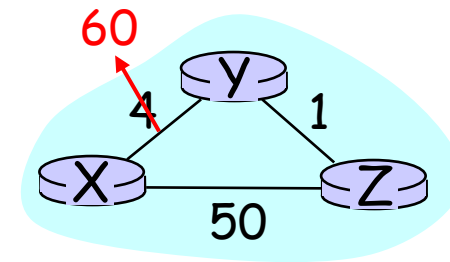


Thuật toán  
Kết thúc

# Distance Vector: Thay đổi giá đường đi

Thay đổi giá đường đi:

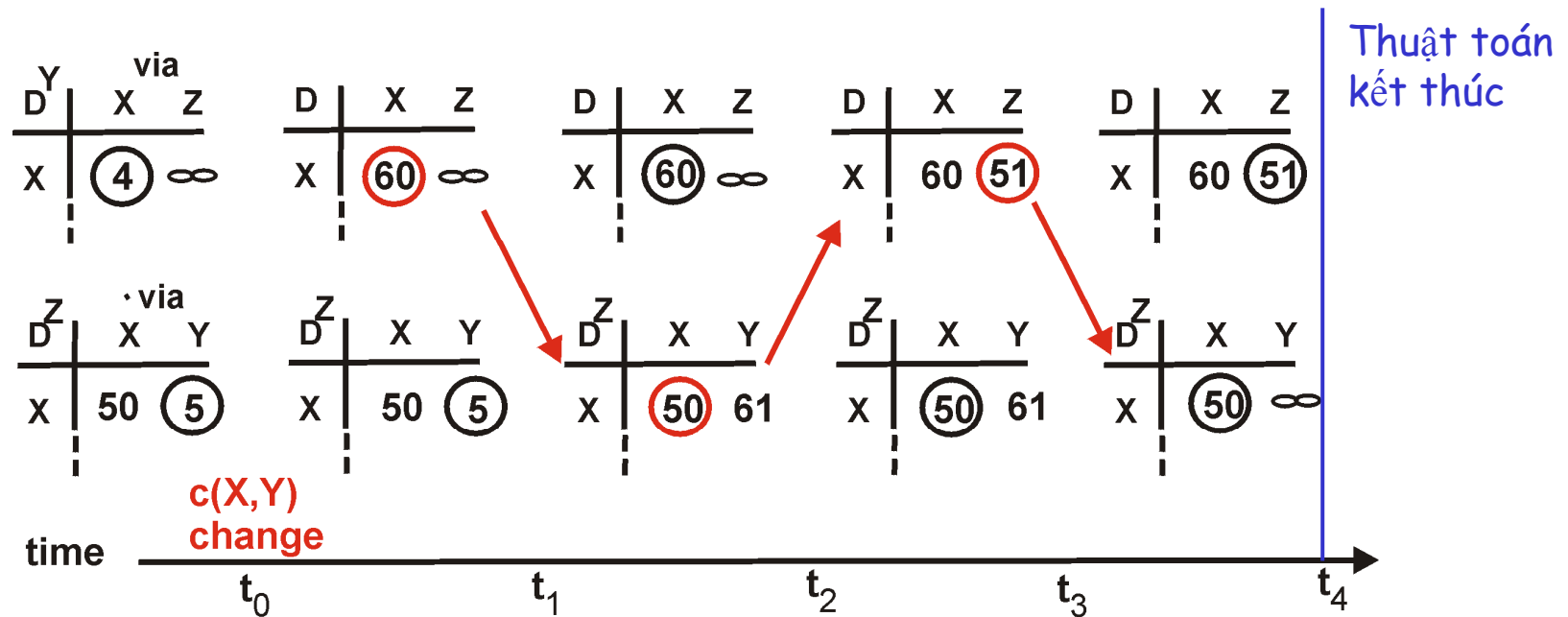
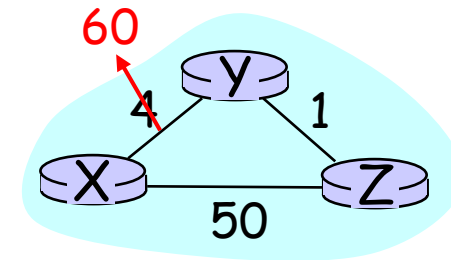
- Tin tốt lan nhanh
- Tin xấu lan lâu !



# Distance Vector: Đảo ngược

Nếu Z định tuyến tới X qua Y:

- Z thông báo với Y khoảng cách từ Z tới X là vô cùng (để Y không chuyển gói tin đến X qua Z)
- Có thể khắc phục vấn đề được không?



## So sánh hai Thuật toán: LS và DV

### Độ phức tạp của Thông điệp

- ❑ LS: Với  $n$  nút,  $E$  links,  $O(nE)$  thông điệp
- ❑ DV: Chỉ trao đổi thông điệp với các Hàng xóm
  - Thời gian Hội tụ biến thiên

### Tốc độ Hội tụ

- ❑ LS: Độ phức tạp  $O(n^2)$  cần  $O(nE)$  thông điệp
  - Có thể bị dao động
- ❑ DV: Thời gian Hội tụ biến thiên
  - Bị vòng lặp
  - Vấn đề count-to-infinity

**Sức mạnh:** Chuyện gì xảy ra nếu router hoạt động không chính xác ?

### LS:

- Nút quảng cáo *giá* sai
- Mỗi nút chỉ tính riêng bảng của mình

### DV:

- Nút có thể quảng cáo tuyến đường sai
- Bảng của nút có thể được các nút khác sử dụng
  - Lỗi lan đi trên toàn mạng

# Định tuyến Phân cấp

Từ trước đến nay định tuyến lý tưởng

- ❑ Tất cả routers giống nhau
- ❑ Mạng “phẳng”

... *Không có* trên thực tế

**Phạm vi:** 50 triệu đích:

- ❑ Không thể lưu trữ 50 triệu địa chỉ trong bảng định tuyến!
- ❑ Khối lượng trao đổi quá lớn!

**Mục tiêu quản trị**

- ❑ Internet = Mạng các Mạng
- ❑ Quản trị viên trong mỗi Mạng muốn giám sát thông tin lưu chuyển trong Mạng của mình



# Định tuyến Phân cấp

- ❑ Sắp xếp router theo vùng, “Miền tự trị” (AS)
- ❑ Router trong cùng AS chạy cùng thuật toán định tuyến
  - Giao thức Định tuyến “nội miền” routing
  - Router trong các AS khác nhau có thể chạy các thuật toán định tuyến nội miền khác nhau

## gateway router

- ❑ Đóng vai trò đặc biệt trong AS
- ❑ Chạy giao thức nội miền để “trò chuyện” với các router khác trong miền
- ❑ Cũng chịu trách nhiệm định tuyến cho các gói tin mà địa chỉ đích ở bên ngoài AS
  - Chạy *thuật toán Định tuyến Liên miền* với các gateway routers khác

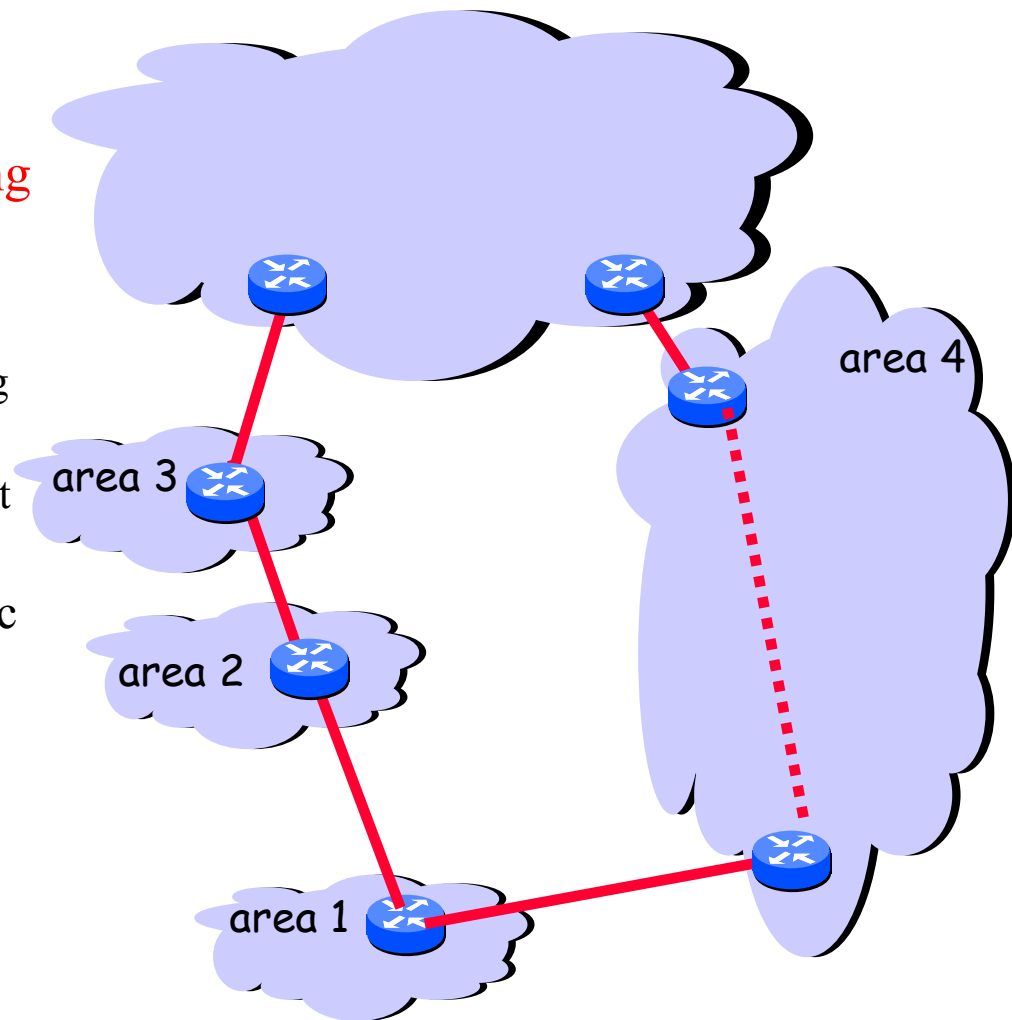
# Tại sao Phải phân loại Nội miền – Liên miền?

## Do phân cấp Định tuyến

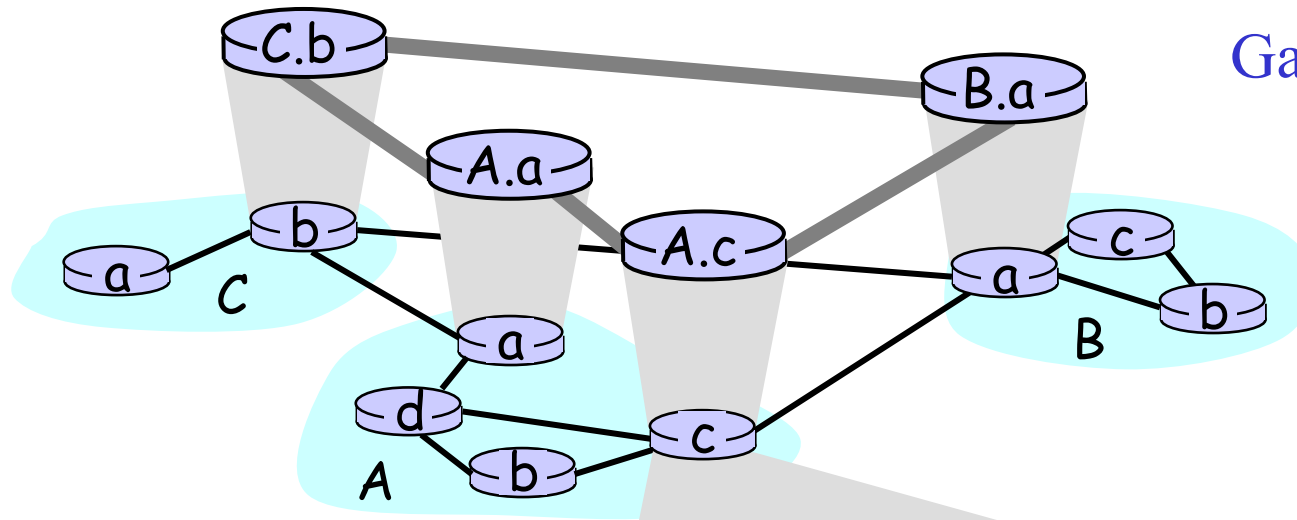
- So sánh với định tuyến phẳng

Sử dụng Định tuyến phân cấp để tăng cường Khả năng mở rộng:

- Với hàng triệu đích đến
  - Không thể lưu trữ tất cả địa chỉ trong Bảng
  - Trao đổi Bảng định tuyến “ngốn” hết băng thông Mạng
- Định tuyến phân cấp giảm kích thước bảng định tuyến và giảm khối lượng trao đổi trên Mạng
  - Vấn đề Chất lượng của Tuyến đường



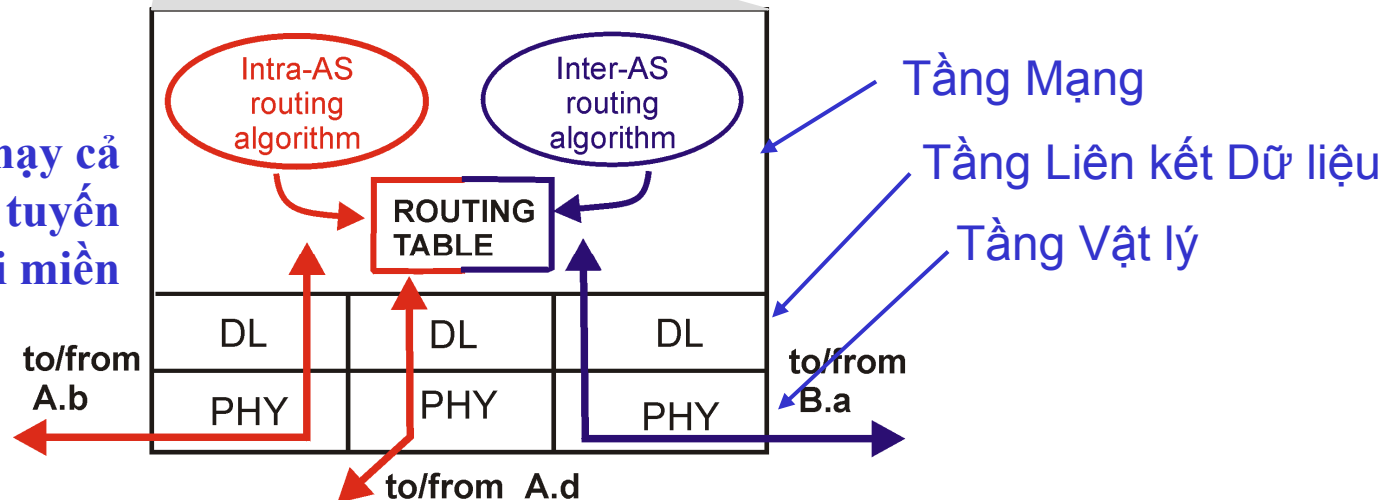
# Định tuyến Nội miền – Liên miền



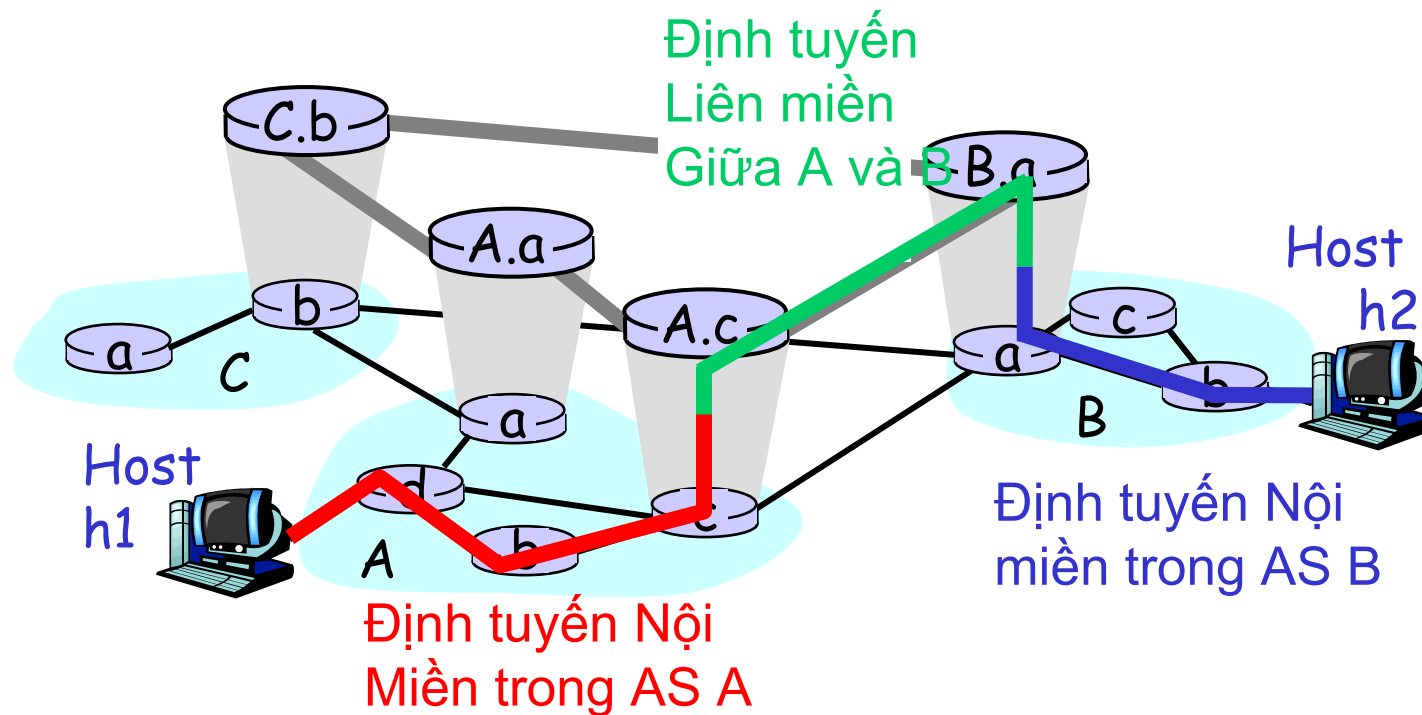
## Gateway:

- Thực hiện định tuyến liên miền
- Thực hiện định tuyến nội miền giữa các AS

gateway A.c chạy cả  
Giao thức Định tuyến  
Liên miền và Nội miền

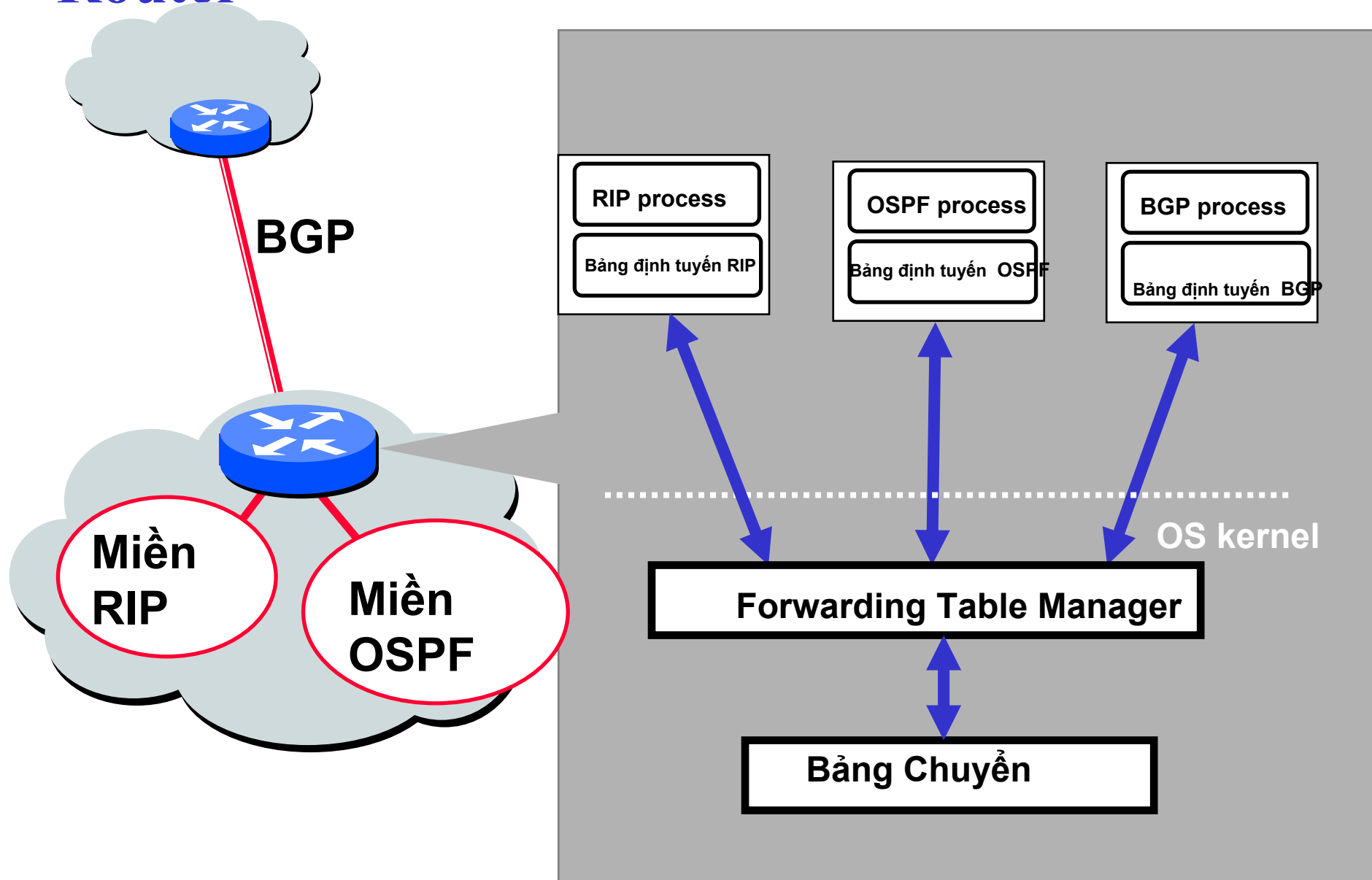


# Định tuyến Nội miền – Liên miền

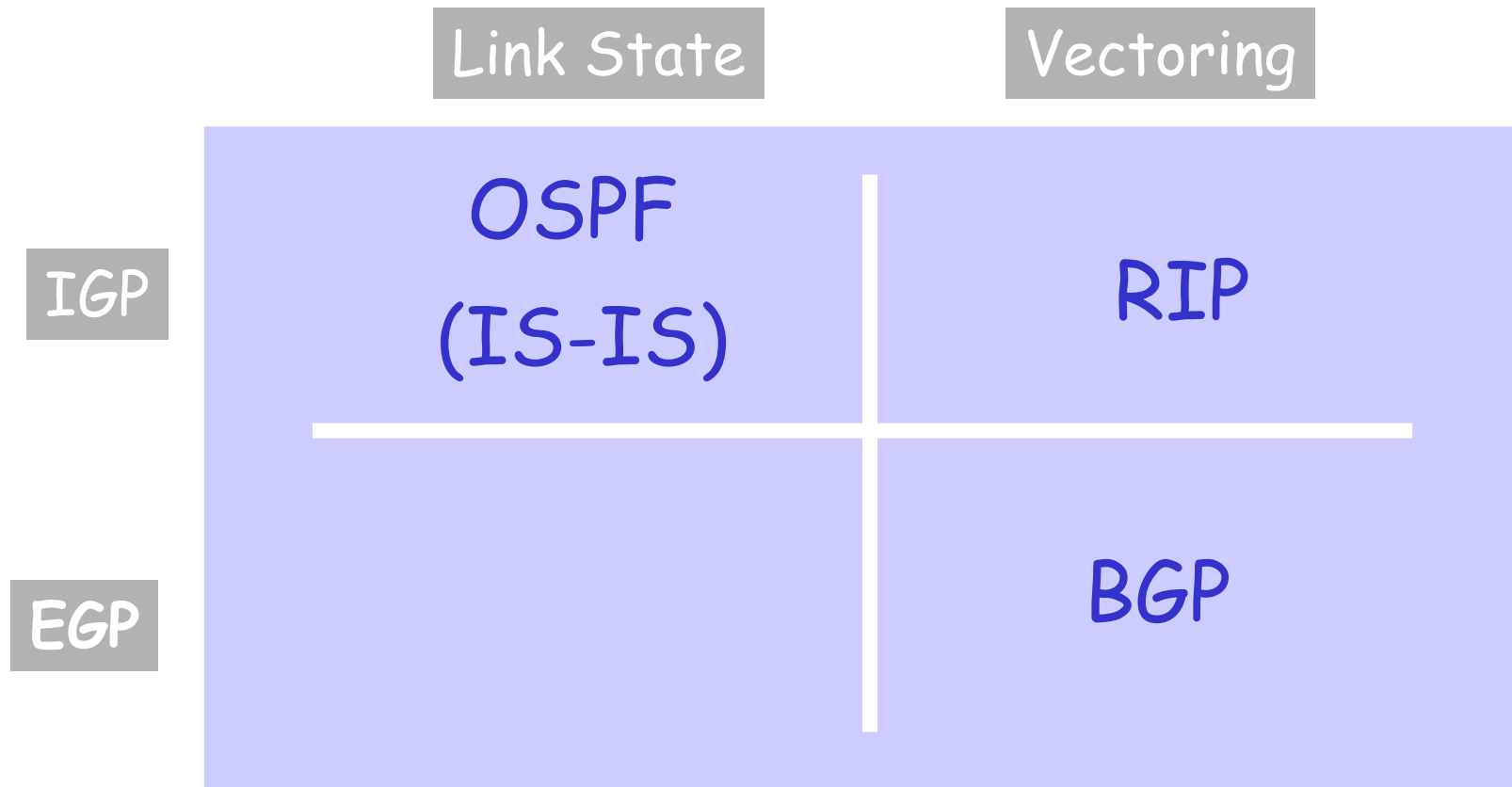


- Chi tiết về các giao thức định tuyến Nội miền và Liên miền sẽ được trình bày trong các phần sau

# Nhiều Tiến trình Định tuyến chạy trên Một Router

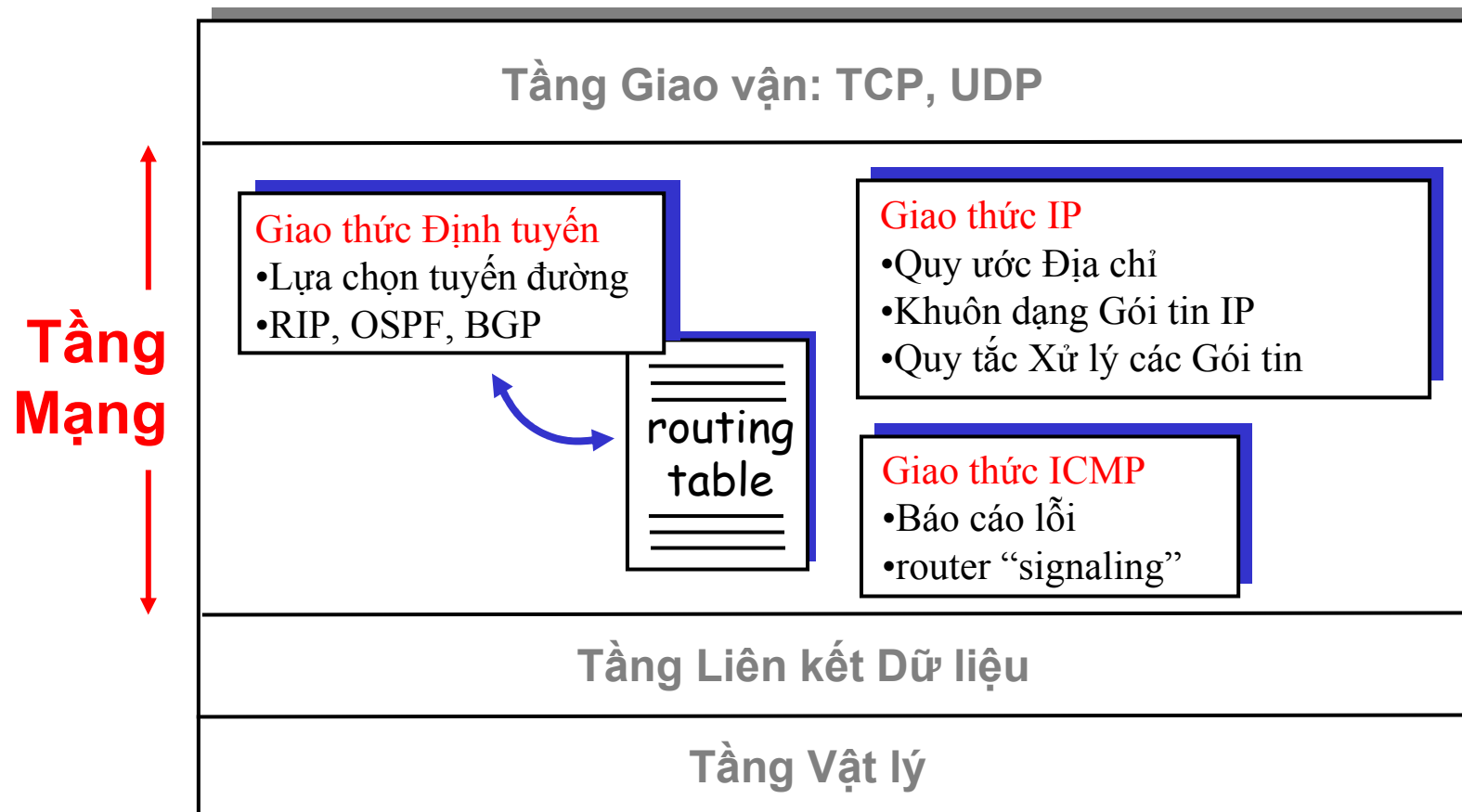


# Bè lũ Bốn Tên 😊



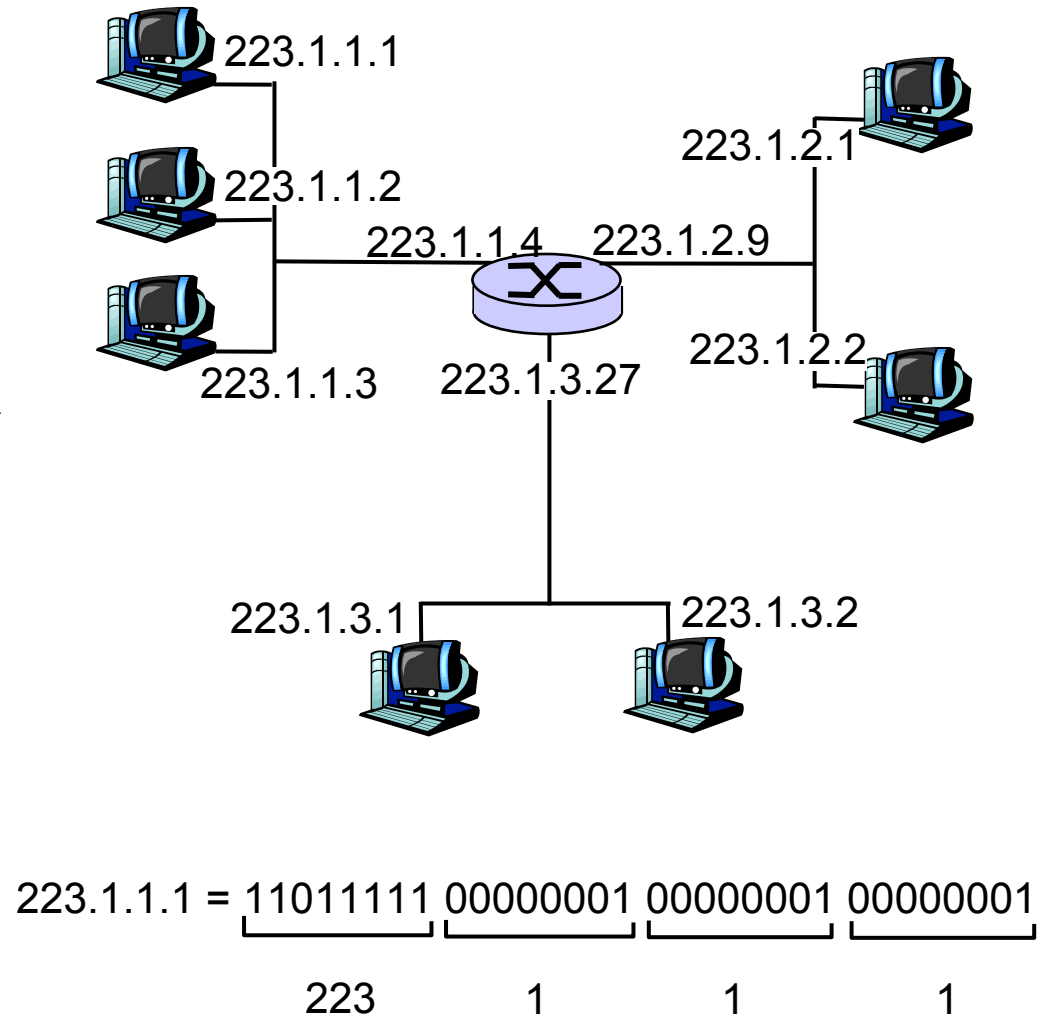
# Tầng Mạng trên Internet

Chức năng của tầng Mạng trên các Máy tính, Router:



# Địa chỉ IP: Giới thiệu

- ❑ Địa chỉ IP : Định danh 32-bit cho *Giao diện* của Máy tính, Router
- ❑ *Giao diện*: Kết nối giữa máy tính, router với kênh truyền Vật lý
  - Router thường có nhiều *Giao diện*
  - Máy tính có thể có nhiều *Giao diện*
  - Địa chỉ IP gắn với giao diện chứ không phải với máy tính hay router





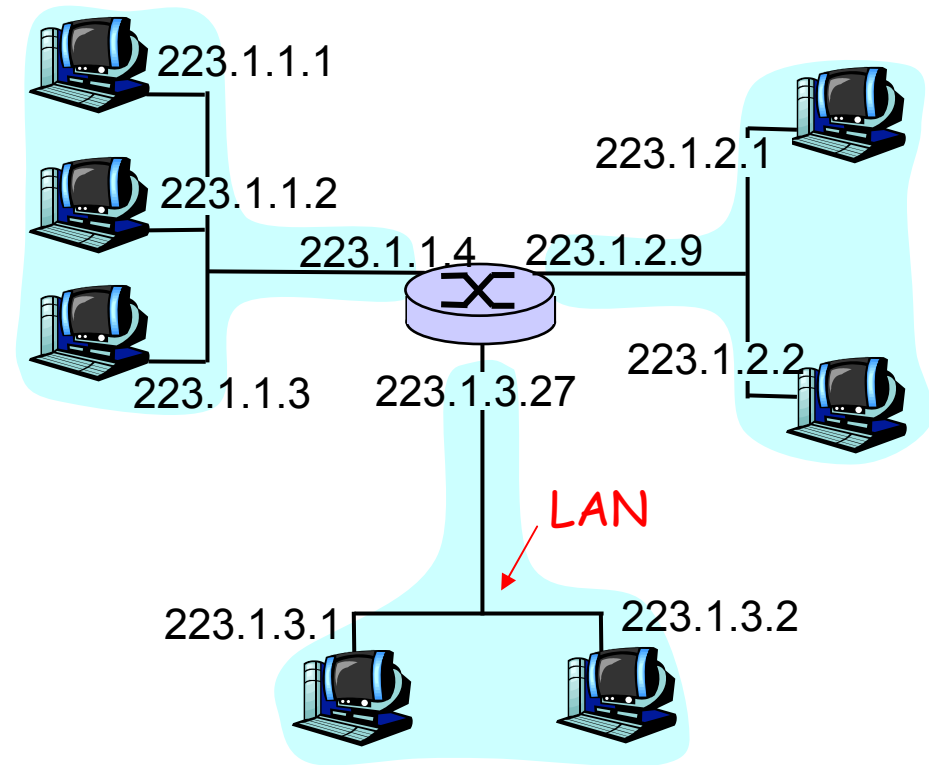
# Địa chỉ IP

## □ Địa chỉ IP:

- Phần network (các bit cao)
- Phần host (các bit thấp)

## □ Thế nào là Mạng ? (Theo quan điểm Địa chỉ IP)

- Giao diện của các thiết bị mà phần network trong địa chỉ IP giống nhau
- Có thể trao đổi dữ liệu với nhau mà không cần qua router



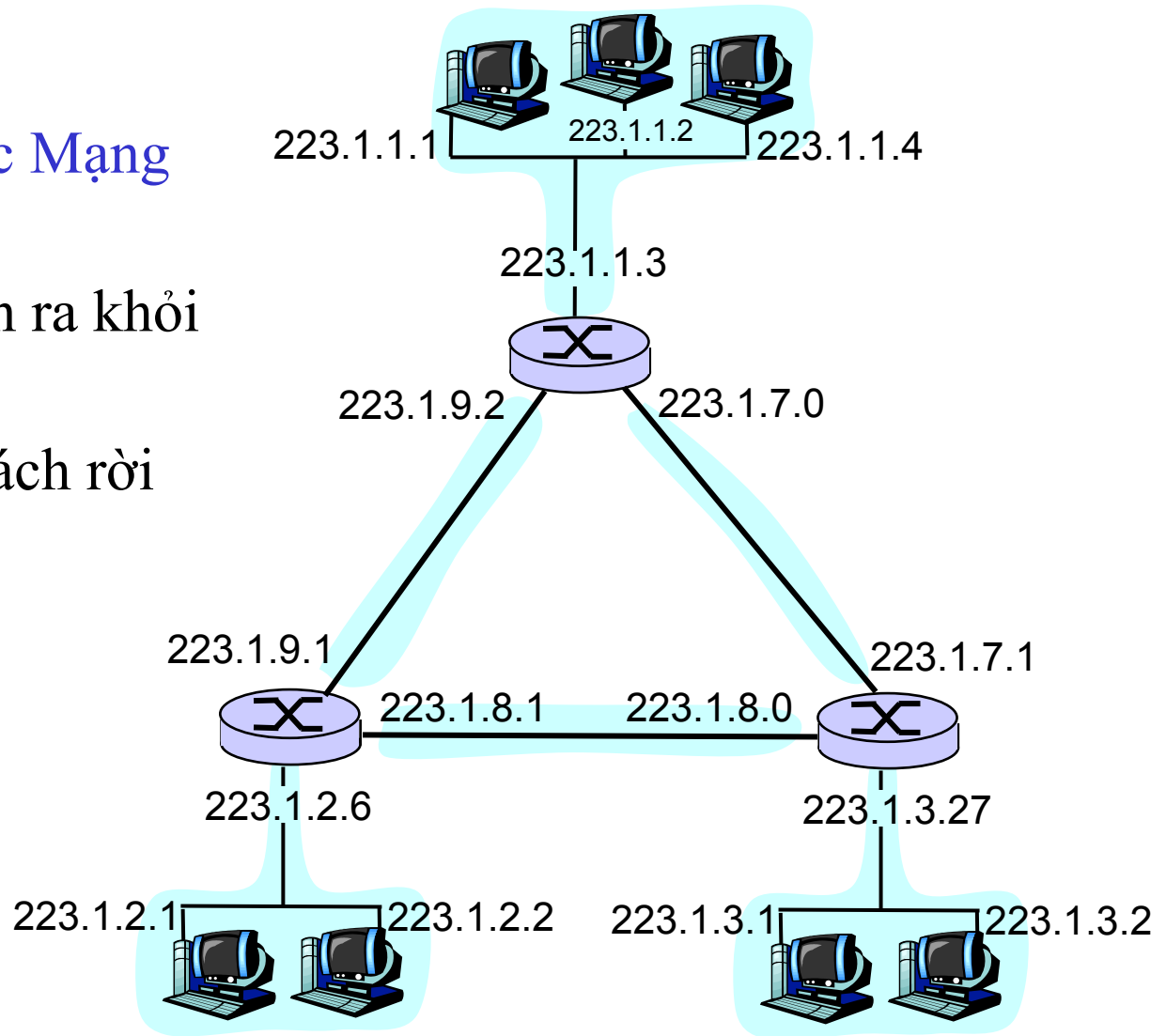
*Mạng với 03 Mạng địa chỉ IP  
(Với địa chỉ IP bắt đầu bằng 223,  
24 bit đầu là phần network)*

# Địa chỉ IP

Làm sao tìm được các Mạng IP?

- ❑ Tách các Giao diện ra khỏi Máy tính, router
- ❑ Tạo ra các Mạng tách rời nhau

**Hệ thống kết nối có 6 mạng**



# Địa chỉ IP

Với định nghĩa mới về “Mạng”, xét lại Địa chỉ IP:

Địa chỉ “phân lớp” :

class

|   |       |         |                   |      |      |                                    |
|---|-------|---------|-------------------|------|------|------------------------------------|
| A | 0     | network |                   | host |      | 1.0.0.0 tới<br>127.255.255.255     |
| B | 10    |         | network           |      | host | 128.0.0.0 tới<br>191.255.255.255   |
| C | 110   |         | network           |      | host | 192.0.0.0 tới<br>223.255.255.255   |
| D | 1110  |         | multicast address |      |      | 224.0.0.0 tới<br>239.255.255.255   |
| E | 11110 |         | for future        |      |      | 240.255.255.255<br>255.255.255.255 |

← 32 bit →

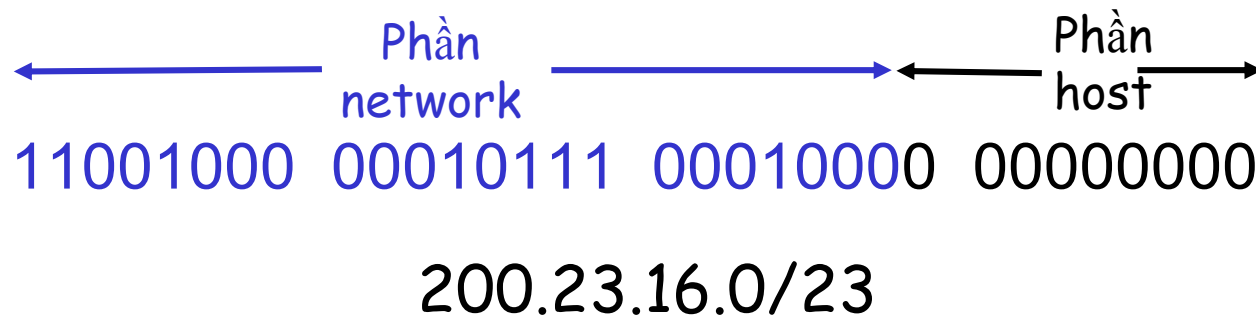
# Địa chỉ IP: CIDR

## ❑ Địa chỉ phân lớp:

- Không gian Địa chỉ bị sử dụng lãng phí và nhanh chóng cạn kiệt
- Ví dụ một lớp B có thể cấp phát tới 65K máy tính, kể cả khi mạng chỉ có 2K máy tính

## ❑ CIDR: Classless InterDomain Routing

- Phần network của địa chỉ có kích thước tùy ý
- Khuôn dạng địa chỉ: **a.b.c.d/x**, trong đó x là số bit trong phần network của địa chỉ



# Địa chỉ IP: Làm sao có được?

Các máy tính (Phần host):

- ❑ Được người quản trị Hệ thống cấu hình cứng và ghi vào file
  - wintel: control-panel->network->configuration->tcp/ip->properties
  - unix:  
%/sbin/ifconfig eth0 inet 192.168.0.10 netmask 255.255.255.0
- ❑ **DHCP: Dynamic Host Configuration Protocol**: tự động xin cấp phát địa chỉ theo kiểu “plug-and-play”
  - Máy tính quảng bá thông điệp “**DHCP discover**”
  - DHCP server trả lời với thông điệp “**DHCP offer**”
  - Máy tính yêu cầu địa chỉ IP bằng thông điệp “**DHCP request**”
  - DHCP server gửi địa chỉ qua thông điệp “**DHCP ack**”

Sử dụng

%whois -h whois.arin.net "n <org>"

Để kiểm tra các địa chỉ cấp phát cho <org>

# Địa chỉ IP: Làm sao có được?

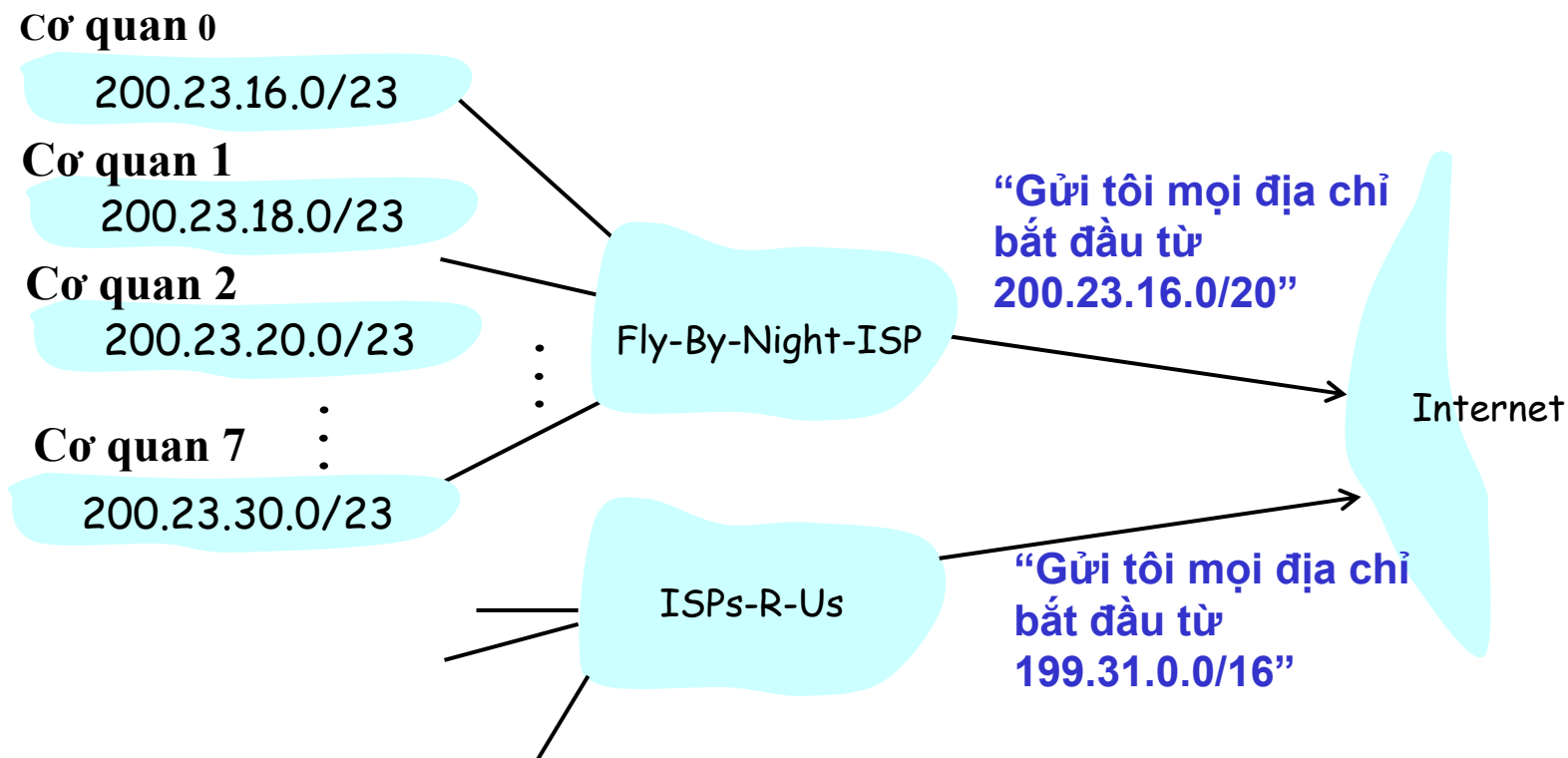
Địa chỉ Mạng (Phần network):

❑ Được cấp phát một phần từ không gian địa chỉ ISP:

|         |   |  |                |
|---------|---|--|----------------|
| ISP     |   | <u>11001000 00010111 00010000</u> 00000000 | 200.23.16.0/20 |
| Cơ quan | 0 | <u>11001000 00010111 00010000</u> 00000000 | 200.23.16.0/23 |
| Cơ quan | 1 | <u>11001000 00010111 00010010</u> 00000000 | 200.23.18.0/23 |
| Cơ quan | 2 | <u>11001000 00010111 00010100</u> 00000000 | 200.23.20.0/23 |
| ...     |   | .....                                      | ....           |
| Cơ quan | 7 | <u>11001000 00010111 00011110</u> 00000000 | 200.23.30.0/23 |

# Địa chỉ Phân cấp: Kết nối các tuyến đường

Định tuyến phân cấp cho phép quảng cáo các thông tin định tuyến một cách **hiệu quả**:



# Địa chỉ phân cấp: Các tuyến cụ thể hơn

ISPs-R-Us có tuyến đường cụ thể tới Cơ quan 1 hơn

**Cơ quan 0**

200.23.16.0/23

**Cơ quan 2**

200.23.20.0/23

**Cơ quan 7**

200.23.30.0/23

**Cơ quan 1**

200.23.18.0/23

Fly-By-Night-ISP

ISPs-R-Us

“Gửi tôi mọi địa chỉ  
bắt đầu từ  
200.23.16.0/20”

“Gửi tôi mọi địa chỉ  
bắt đầu từ 199.31.0.0/16  
hoặc 200.23.18.0/23”

Internet



## Địa chỉ IP: cuối cùng...

Q: Làm sao ISP có được một dải địa chỉ?

A: **ICANN**: Internet **C**orporation for **A**ssigned  
**N**ames and **N**umbers

- Cấp phát Địa chỉ
- Quản lý DNS
- Gán tên miền, quản lý tranh chấp

# Đề gửi datagram từ Nguồn đến Đích

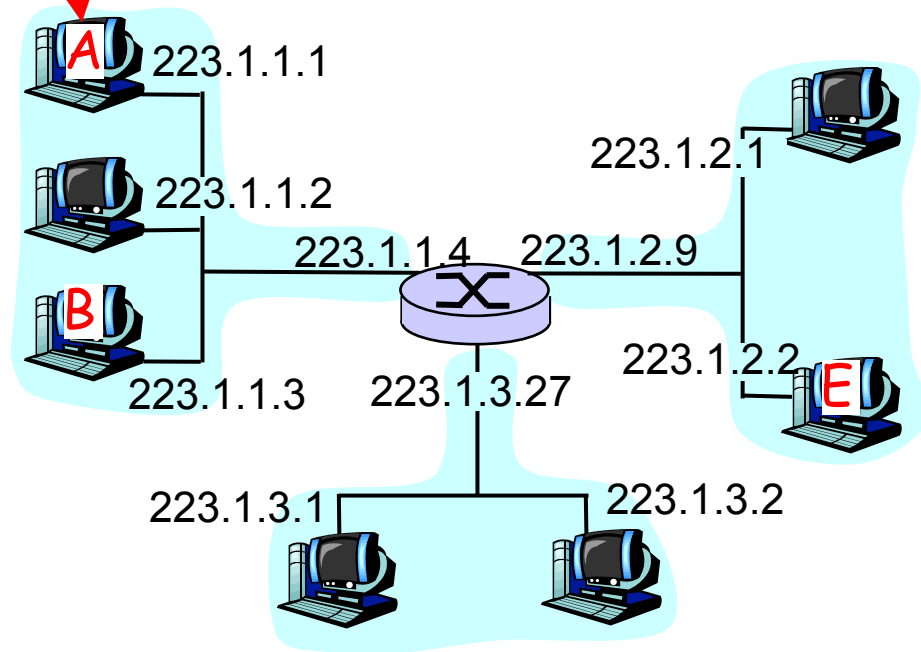
IP datagram:

| misc<br>fields | source<br>IP addr | dest<br>IP addr | data |
|----------------|-------------------|-----------------|------|
|----------------|-------------------|-----------------|------|

- ❑ datagram gần như không thay đổi trên tuyến đường từ đích đến nguồn
- ❑ Chỉ quan tâm tới địa chỉ nhận

**Bảng Định tuyến ở A**

| Dest. Net. | next router | Nhops |
|------------|-------------|-------|
| 223.1.1    |             | 1     |
| 223.1.2    | 223.1.1.4   | 2     |
| 223.1.3    | 223.1.1.4   | 2     |

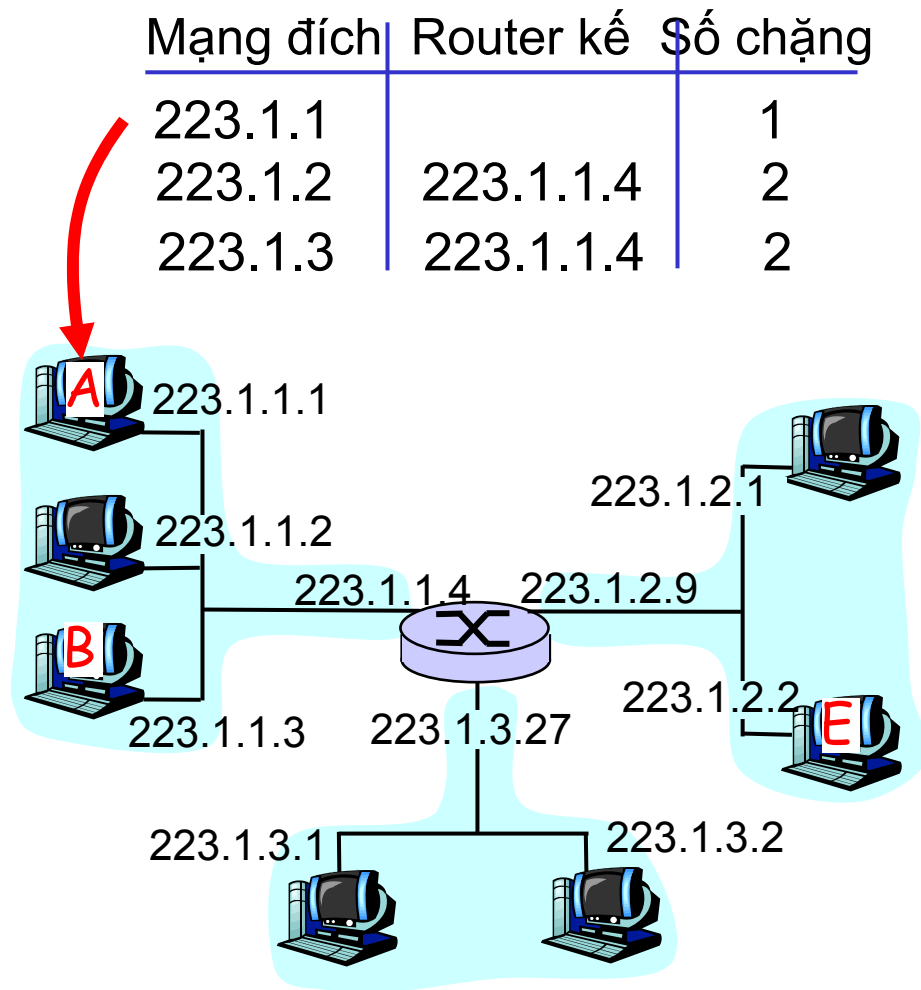


# Đề gửi datagram từ Nguồn đến Đích

|             |           |           |      |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.1.3 | data |
|-------------|-----------|-----------|------|

**Bắt đầu từ A, với gói tin IP có địa chỉ đích là B:**

- ❑ Tìm kiếm địa chỉ B trong bảng
- ❑ Thấy rằng B nằm trong cùng Mạng với A
- ❑ Tầng liên kết dữ liệu chịu trách nhiệm chuyển IP datagram trực tiếp tới B bên trong frame của tầng liên kết
  - B và A trực tiếp kết nối với nhau

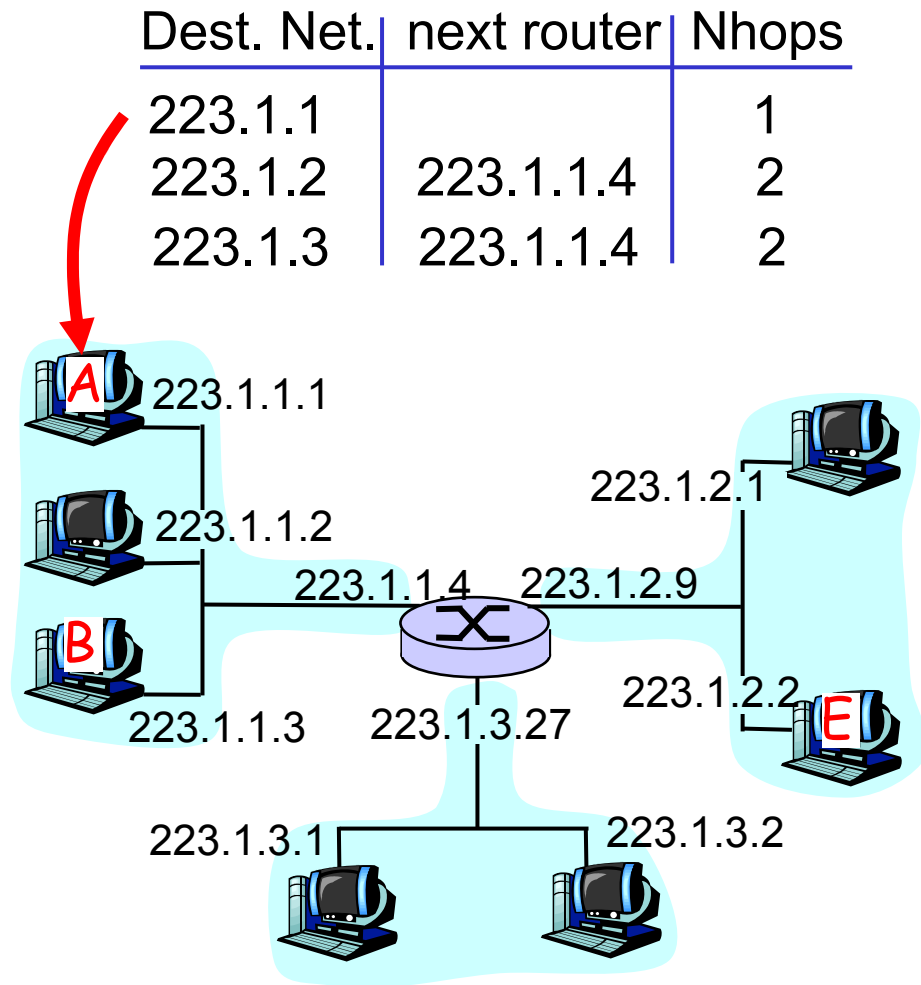


# Đề gửi datagram từ Nguồn đến Đích

|             |           |           |      |
|-------------|-----------|-----------|------|
| misc fields | 223.1.1.1 | 223.1.2.3 | data |
|-------------|-----------|-----------|------|

Bắt đầu từ A, đích là E:

- ❑ Tìm kiếm địa chỉ E trong mạng
- ❑ E ở mạng *khác*
  - A, E không có kết nối trực tiếp
- ❑ Trong bảng định tuyến: router kế tiếp tới E là 223.1.1.4
- ❑ Tầng liên kết dữ liệu gửi datagram tới router 223.1.1.4 bên trong frame của tầng liên kết
- ❑ datagram đến 223.1.1.4
- ❑ *Tiếp tục.....*



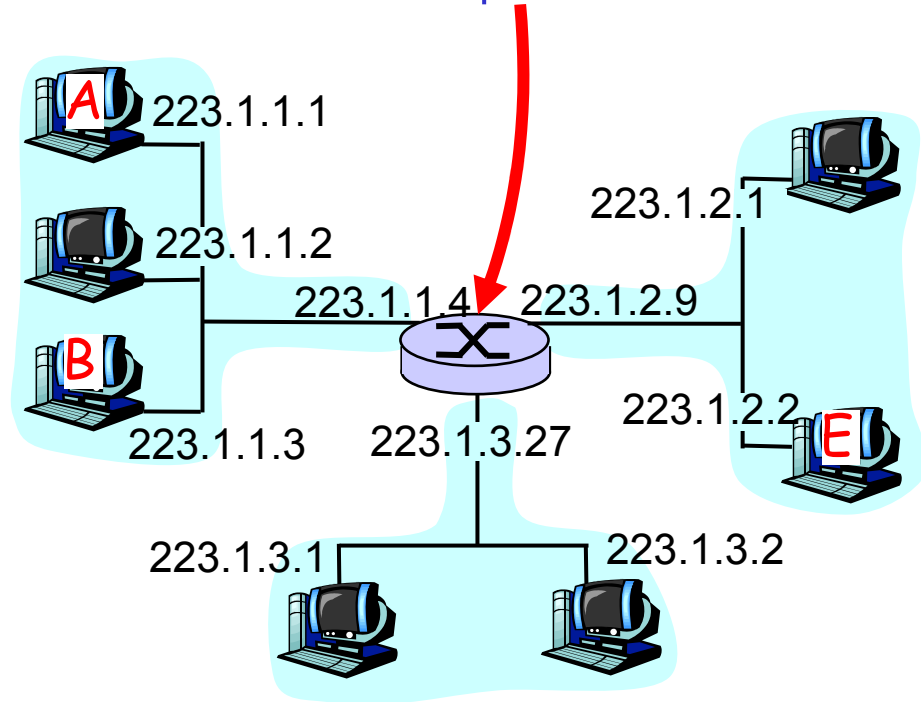
# Đề gửi datagram từ Nguồn đến Đích

| misc fields | 223.1.1.1 | 223.1.2.3 | data |
|-------------|-----------|-----------|------|
|-------------|-----------|-----------|------|

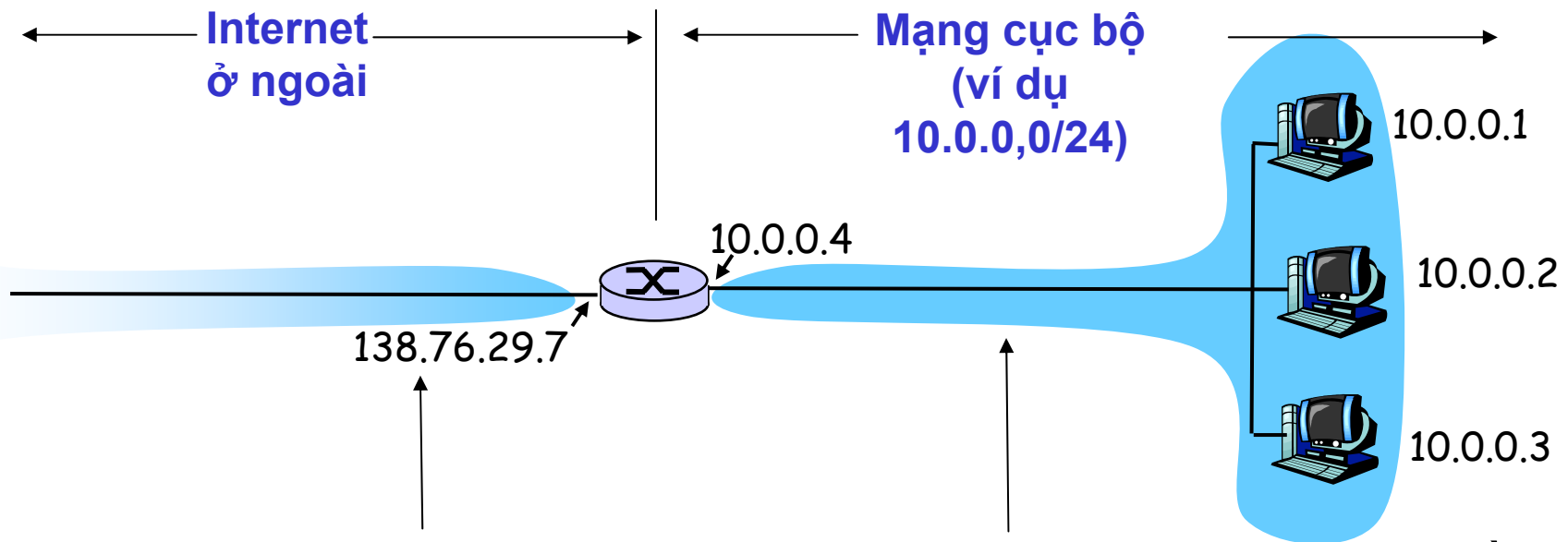
Đến 223.1.4, đích là 223.1.2.2

- ❑ Xác định địa chỉ đích là E
- ❑ E trên cùng mạng với giao diện 223.1.2.9 của router
  - router, E có kết nối trực tiếp
- ❑ Tầng liên kết dữ liệu gửi datagram tới 223.1.2.2 bên trong frame tầng liên kết dữ liệu qua giao diện 223.1.2.9
- ❑ Gói tin đã đến được 223.1.2.2!!!  
(*Chúc mừng!*)

| Mạng đích | router kế tiếp | Số chặng | Giao diện  |
|-----------|----------------|----------|------------|
| 223.1.1   | -              | 1        | 223.1.1.4  |
| 223.1.2   | -              | 1        | 223.1.2.9  |
| 223.1.3   | -              | 1        | 223.1.3.27 |



# NAT: Network Address Translation



*Tất cả* datagram *rời* mạng cục bộ có *cùng* địa chỉ IP đích (đã bị biến đổi):  
138.76.29.7,  
Tuy nhiên cổng nhận giá trị khác nhau

Datagram với địa chỉ đích/ nguồn ở trong mạng IP (địa chỉ 10.0.0/24) sẽ giữ nguyên địa chỉ (như bình thường)

# NAT: Network Address Translation

- ❑ **Động lực:** Mạng cục bộ chỉ có một địa chỉ IP để kết nối với bên ngoài:
  - Không cần thiết phải có 1 khoảng địa chỉ IP từ ISP: - một địa chỉ IP được sử dụng chung cho tất cả các thiết bị
  - Bên ngoài không nhìn thấy sự thay đổi địa chỉ bên trong
  - Có thể thay đổi ISP mà không thay đổi IP của các máy tính bên trong
  - Bên ngoài không thể nhìn thấy địa chỉ tường minh của các thiết bị bên trong mạng cục bộ.

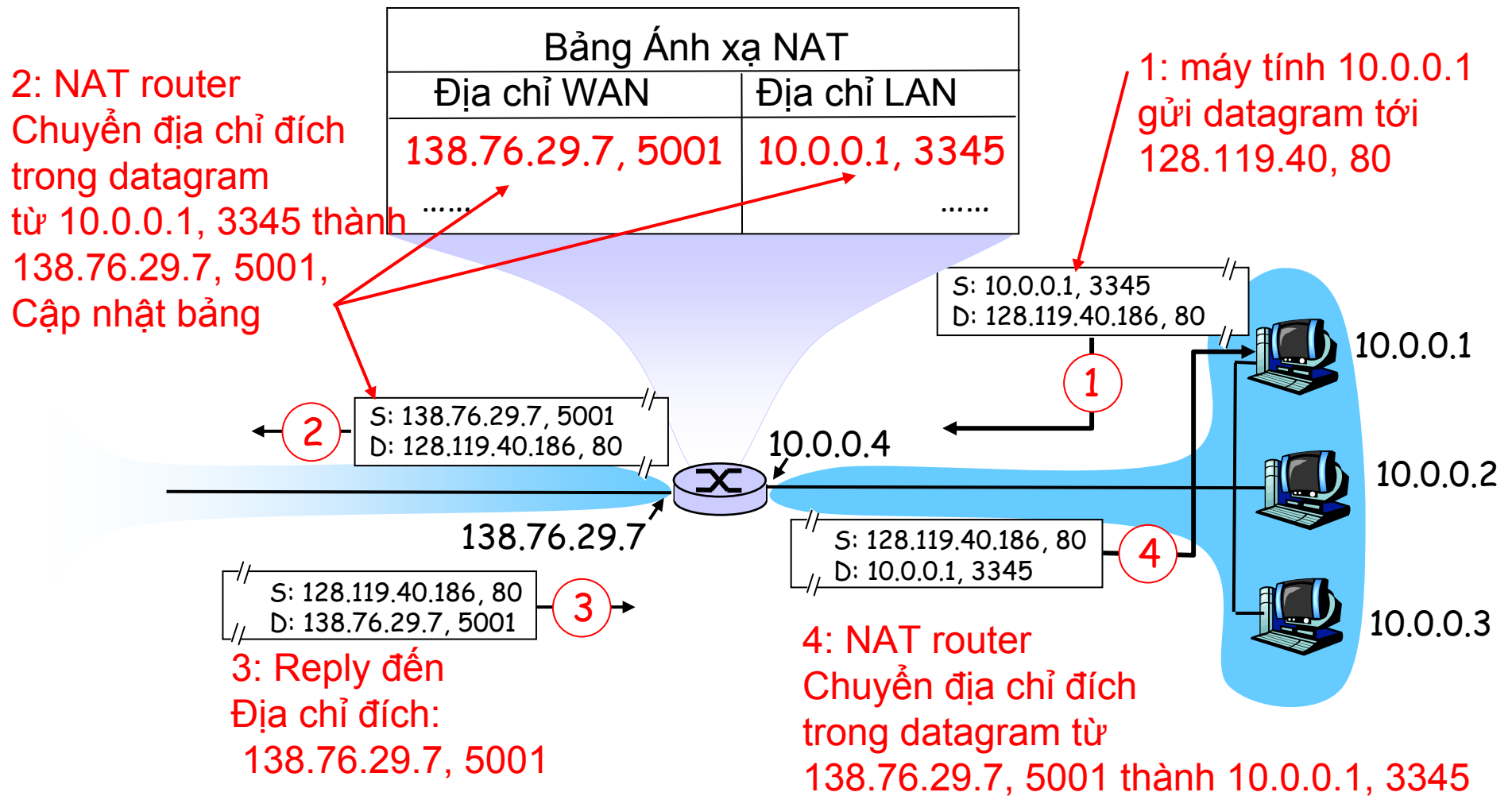
# NAT: Network Address Translation

**Cài đặt:** NAT router phải:

- *Datagram chuyển ra ngoài: thay thế* (IP gửi, port #) tất cả datagram chuyển ra ngoài thành (địa chỉ IP được chuyển, cổng # mới)  
... Tương tác clients/servers sử dụng (địa chỉ IP được chuyển, cổng # mới) làm địa chỉ đích.
- *Ghi nhớ (trong bảng biến đổi địa chỉ)* tất cả các cặp (IP đích, port #) thành (địa chỉ IP được chuyển, port # mới) phục vụ cho mục đích chuyển đổi
- *Datagram đến: thay thế* (địa chỉ IP được chuyển, port # mới) trong các trường địa chỉ đích của mỗi datagram đến với ánh xạ tương ứng (IP gửi, port #) lưu trong bảng NAT



# NAT: Network Address Translation



# NAT: Network Address Translation

## ❑ 16-bit địa chỉ cổng:

- 60,000 kết nối đồng thời trên cùng một địa chỉ mạng LAN duy nhất !

## ❑ NAT có vấn đề:

- Router chỉ nên xử lý ở tầng 3
- Vi phạm nguyên tắc đầu cuối
  - Các nhà phát triển ứng dụng, đặc biệt các ứng dụng P2P phải tính toán đến việc sử dụng NAT
- Có thể việc khan hiếm địa chỉ sẽ hết khi sử dụng IPv6